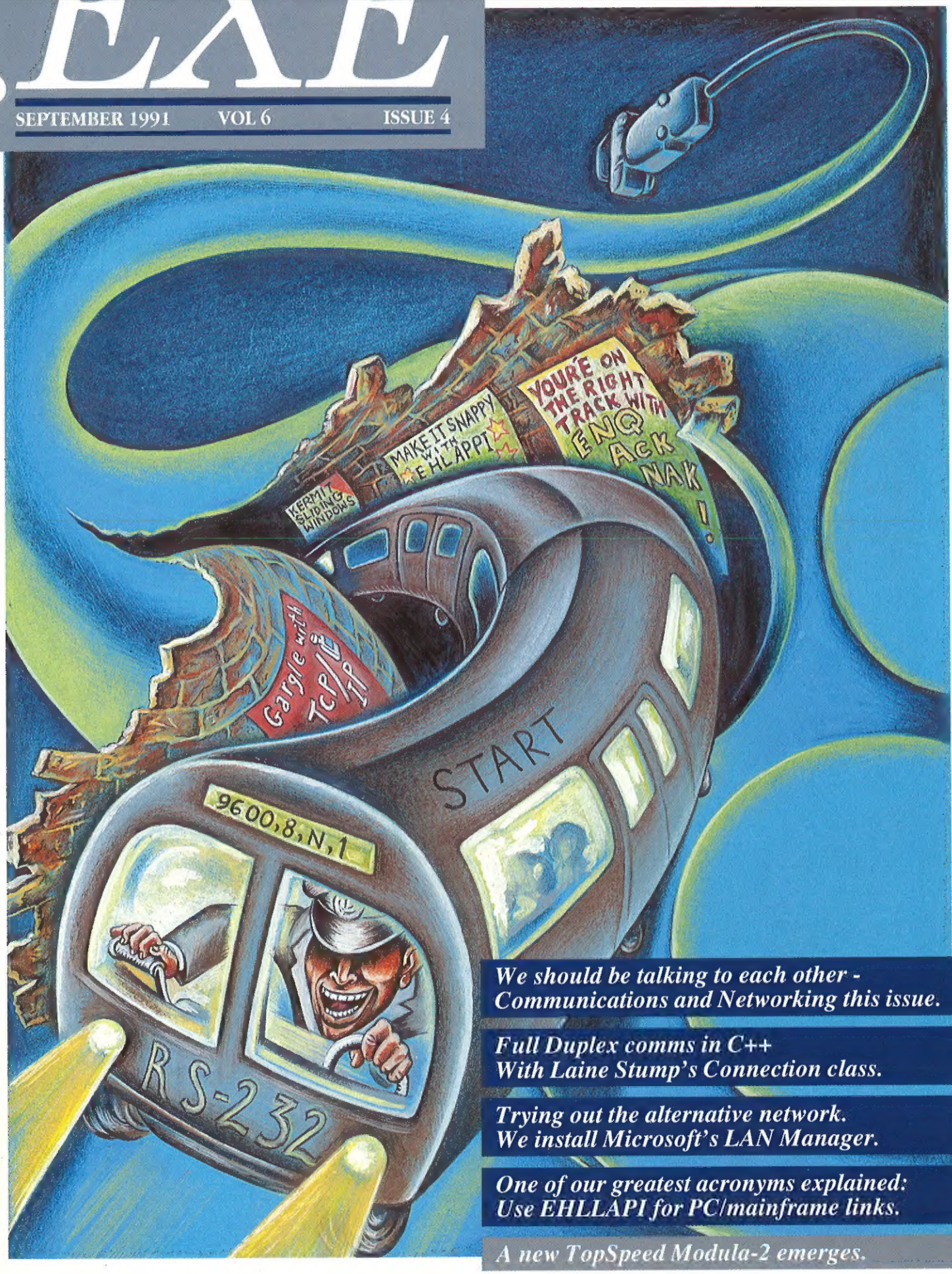
# EXE

SEPTEMBER 1991 VOL 6 ISSUE 4

£3.20

*The Software Developers' Magazine*




*We should be talking to each other - Communications and Networking this issue.*

*Full Duplex comms in C++ With Laine Stump's Connection class.*

*Trying out the alternative network. We install Microsoft's LAN Manager.*

*One of our greatest acronyms explained: Use EHLLAPI for PC/mainframe links.*

*A new TopSpeed Modula-2 emerges.*

# *Runs on everything... well, almost*

***Glockenspiel C++ Compilers for UNIX***

Glockenspiel C++ runs on the widest range of machines, from 386 PC's to the largest VAX. With over 120 supported machine/operating system configurations, including DEC, RS6000, Suns, SCO and many others, chances are we have the one you want. All Glockenspiel C++ Compilers are AT&T C++ conformant and all Glockenspiel C++ code is consistent across platforms

***Glockenspiel CommonView for Motif***

CommonView is the standard class library for the rapid creation of Windows and OS/2 PM Graphical User Interfaces in C++. Now CommonView is available for Motif, making it easy to write Motif applications and making those applications portable to the PC.

For full details call The Products Group on 0285 - 655888

*glockenspiel* C++

CIRCLE NO. 148

QA

Please send more information on:

☐ Glockenspiel C++ for UNIX ☐ CommonView Class Libraries for Motif, Windows or PM
☐ C++ for PC's ☐ Training ☐ Consultancy

NAME TITLE

COMPANY

ADDRESS

TEL. NO.

Mail to: The Products Group, QA Training Ltd., Cecily Hill Castle, Cirencester, Gloucestershire GL7 2EF. UK.
Tel: 0285 655888 Fax: 0285 640181

Vol 6 Issue 4

# .EXE

September 1991

**Editor:** Willie Watts
**Staff Writer:** Paul Kemp
**Trainee Staff Writer:** Cliff Saran
**Production Manager:** Katerina Adams
**Design & Layout:** Mark English
**Group Advertising Manager:** Sandra Inniss-Palmer
**Assistant Advertising Manager:** Ed Butcher
**Advertising Executives:** Chris Haill, Marc Warren
**Reader Services & Promotions:** Helena Adams
**Publisher:** Jon Howell
**Administration & Subscriptions:** Rena Gibbs, Sandy McDonnell
**Front Cover Illustration:** Mike Nicholson
**(Yes, we know we spelled EHLLAPI wrong!)**

**General**
.EXE Magazine is independent and not affiliated to any vendor of hardware, software or services. It is published by Process Communications Ltd, 10 Barley Mow Passage, Chiswick, London W4 4PH.
Tel: (Advertising/Editorial/Production) 081 994 6477
(Subscriptions) 044282 4501
Facsimile: 081 994 1533 Telex: 8811418 SPACES G
ISSN: 0268-6872

**Subscriptions**
.EXE Magazine is a monthly journal for software developers. It is available only by subscription, at a cost of £35 per annum (11 issues) in the UK. 'A subscription implies that this journal will be sent to the subscriber until one of the three expires' - AG Macdonell. The magazine is published around the 26th of the month preceding the cover date. There is no January issue. To subscribe or obtain details of overseas/academic rates, please call 044282 4501, or write to The Subscriptions Manager, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We can invoice your company or take ACCESS and VISA cards. In case of a query about your subscription, please call 044282 4501.
Back issues are available at a cost of £3.50 each - call our Chiswick office (081 994 6477) for a list of issues still in print. We also offer a range of magazine binders, disk holders etc - please call for details.

**Editorial**
Editorial enquiries should be addressed to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We welcome letters, opinions, suggestions and articles from our readers. Please write for a copy of our Contributors' Guide.
Information contained in .EXE is believed to be correct. If errors are found, we will endeavour to publish a clarification in the next available issue.
From time to time, we offer to copy the PC program code described in an article onto diskette for our readers. In this case, please send a blank, formatted disk with a self-addressed, prepaid mailer to the editorial address given above. We can copy both 5.25" and 3.5" disks.
The publishers can accept no liability for any consequences of using software distributed in this way.

**Advertising**
If you are interested in advertising in this magazine, please write to the Display Advertising Manager/Recruitment Sales Manager (as appropriate), .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH, or call 081 994 6477 for details of our advertising rates.

**Pronunciation**
The name of .EXE Magazine is pronounced to rhyme with 'not sexy magazine'.



***Issue theme: Communications and Networking***

# I was a teenage user

*Everybody loves to hate the user. If there is a bug in the software, if the PC breaks down or if someone goes sick with the 'flu, blame the users! Tracy-Anne Ormrod wonders why.*

You want to arrange a meeting with the user. You could phone, but you are bored at the moment and could do with some exercise. After locating the department, you endeavour to locate your user. She is nowhere to be seen, so you sit down at a desk (barely visible under piles of paper, Garfields and pictures of men in tight swimming trunks) and construct an image of your potential enemy. She appears, and confirms your worst (/best?) fears. Medium height, blonde, slim, fashionably short skirt and long painted finger nails. You approach nervously, introduce yourself, and set a date for a meeting.


Cut to that day. You arrive in the department to be greeted with a smile and a cup of coffee. You haven't prepared - why should you? You are the expert. You hit her with your first question, a blinder this, 'What colours would you like the screens?'. This is greeted with a perplexed look. This meeting hasn't got off to a good start. Try Plan B: 'What is your job function, and what part of it is to be automated?'

An hour later you emerge drained but mostly intact, apart from having conceded a promise to produce a specification. Returning to the IS department, your manager informs you of the deadline: no time for a proper spec, pull your finger out and get coding.

Two weeks later. While searching for a particularly elusive bug in a validation routine the phone rings - it's La User. She wants the specification. You explain the lack of time. She still wants the specification. You retort that you know how to do your job, you understand your priorities, thank you very much. She snorts and slams down the phone. You return to your debugging, smarting from the encounter. Who does she think she is?

The day of reckoning. You have improvised a system, based on your scant knowledge of the business requirements. You are greeted by a sterner looking user than last time but a different desk. It is neat, tidy, not a piece of paper in sight except for a few Post-it notes (arranged in prioritised order). Ah. It turns out that this is her *real* desk, and not the Garfield/semi-nude men heap. Gloom engulfs you, and you begin to pray that the bug you fixed late last night doesn't appear again...

Users do have brains - admittedly some use them better than others, but that is also true of people in computing departments. Users also invariably know an incredible amount about the business and are much more educated about the use and power that computers have than we give them credit for. When I was a user, what used to annoy me most about the so-called computer expert was the patronising way he would talk to you. I know how it feels to be treated like an idiot.

My parable serves to illustrate some of the common faults that we programmer types exhibit when interviewing our users. First of all, lack of preparation for the meeting. Always map out an agenda. If you have no documentation about the proposed system ask the user for some before your meeting.

Don't wander round to their department to arrange the meeting unless strictly necessary. Users have work to do as well.

Take a tape recorder with you for the meeting to record the vital information. Ask 'open' questions first of all, to help establish a rapport and give you general information. Save the specific questions until the end, and make sure you get specific answers. Use plain English, not jargon.

Graphical techniques, such as Data flow diagrams and Data model or Entity relationship diagrams, depict the system much better than a 'Victorian novel' specification. They are much quicker, too. And yes, the users *can* understand what is happening in the diagrams, if you deign to explain the notation. If you bung a standard explanation of the notation in the spec, and give it to the users *before* your meeting, they will have sensible and often critical questions for you on the day. On projects with a team of developers, introduce design walk-through meetings with a user representative in attendance to give his point of view. This builds up a good relationship and helps the users develop realistic expectations.

What this all adds up to is treating users as a human beings, with their own valid points of view and with equal rights to the system you are constructing; plus making sure that you as an individual conduct yourself professionally. Respect their knowledge, and remember you are working together for the success of the system. You are the expert in the construction of the system, so if anything goes wrong it is your fault and not the users!

I hope with this advice you will get the most from your user. And remember: One Garfield doesn't make a bimbo.

EXE

---

*Tracy-Anne Ormrod is a director of Applications Technology Ltd. She can be contacted on 0491 35187 for controversial opinions on the state of systems development today. The author assures us that her description of a user is not a self-portrait.*

# GREY MATTER

*Prigg Meadow, Ashburton, Devon TQ13 7DF*

Prices do not include VAT or other local taxes, but do include delivery in the UK and Europe. Please check prices at time of order as ads are prepared some weeks before publication. This page lists some of our products– call us for a complete price list and details of discount structure. ORDER BY PHONE WITH YOUR CREDIT CARD.

**TEL: (0364) 53499**

FAX: (0364) 53071

## C++ COMPILERS

| | | |
|---|---|---|
| Topspeed C++ Std | OS/2 | £110 |
| Topspeed C++ Prof | S/2 | £165 |
| Zortech C++ Dev v3 | DOS&OS/2 | £360 |
| Zortech C++ Sci&Eng | DOS&OS/2 | £540 |
| Borland C++ | PC-DOS | £206 |
| Guidelines C++ | PC-DOS | £370 |
| Topspeed C++ Std | C-DOS | £110 |
| Topspeed C++ Prof | PC-DOS | £165 |
| Turbo C++ | PC-DOS | £ 50 |
| Zortech C++ v3 | PC-DOS | £225 |

### LIBRARIES & SUNDRIES

| | | |
|---|---|---|
| C++/Views (ZC++) | WIN | £270 |
| Zinc Interface Lib (C++) | PC-DOS | £155 |
| Zinc Interface Lib | WIN&DOS | £230 |
| Zortech C++ Database | PC-DOS | £180 |
| Win++ | WIN | £160 |
| Codebase++ | WIN&DOS | £195 |

## PROGRAMMING TOOLS

| | |
|---|---|
| **Ada Compilers** | **Algol Compilers** |
| **Assemblers & Libs** | **AWK** |
| **Basic Compilers** | **Basic Interpreters** |
| **Basic Utilities** | **Basic Libraries** |
| **C Compilers** | **C Utilities** |
| **C Interpreters** | **C Libraries** |
| **C++** | **Cobol Compilers** |
| **Comms.Libraries** | **Cross Assemblers** |
| **Database Libs.** | **Debuggers** |
| **Dis-assemblers** | **Editors** |
| **Engineers Libs.** | **Expert Systems** |
| **Forth** | **Fortran Compilers** |
| **Fortran Libraries** | **Graphics Libraries** |
| **Icon** | **Linkers/Locaters** |
| **Lisp** | **Logo** |
| **Modula-2** | **Nial Interpreters** |
| **Object Oriented** | **OPS 5** |
| **Pascal Compilers** | **Pascal Libraries** |
| **Prolog** | **Rexx** |
| **Risc** | **Screen Libraries** |
| **Smalltalk** | **Snobol** |

**We stock many items for which there is no space in these advertisements.**

## THE C LANGUAGE

*Refer also to the C++ section.*

### C COMPILERS

| | | |
|---|---|---|
| Microsoft C v6 | OS/2&PC-DOS | £222 |
| Topspeed C for OS/2 | OS/2 | £335 |
| Watcom C v8 Prof | OS/2&DOS | £325 |
| Aztec C86 Professional | MS-DOS | £120 |
| Aztec C86 Developer | MS-DOS | £175 |
| Aztec C86 Commercial | S-DOS | £275 |
| HIGH C v1.61 | MS-DOS | £390 |
| HIGH C 386 Global | PL386&MS-DOS | £650 |
| Hi-Tech C | MS-DOS | £165 |
| Hi-Tech C 386 PDS | PC-DOS | £385 |
| Microsoft QuickC v2.5 | PC-DOS | £ 50 |
| M'soft QuickC/QuickAsm | PC-DOS | £ 83 |
| MIX Power C & lib s'ce | C-DOS | £ 30 |
| Topspeed C Std | PC-DOS | £110 |
| Topspeed C Prof | PC-DOS | £165 |
| Watcom C v8 Std | MS-DOS | £260 |
| Watcom C/386 Std | PL386&MS-DOS | £585 |
| Watcom C/386 Prof | PL386&MS-DOS | £780 |
| Aztec C Personal 1.06D | CP/M-80 | £120 |
| Aztec Commercial 1.06D | CP/M-80 | £200 |
| Hi-Tech C | Z80 + CP/M-80 | £105 |
| Mix C | 80 + CP/M-80 | £ 35 |
| Aztec C68K MPW C | MACINTOSH | £110 |
| Aztec C68K C | MACINTOSH | £ 75 |
| Aztec C68K/AM Developer | AMIGA | £175 |
| Aztec C68K/AM Prof'snl | AMIGA | £120 |
| Aztec C68K/ST Developer | ATARI | £175 |
| Aztec C68K/ST Prof'snl | ATARI | £120 |
| Hi-Tech C 68000 | ATARI | £140 |
| Mark Williams C v2 | ATARI | £110 |
| Prospero C | ATARI | £ 55 |

### C INTERPRETERS

| | | |
|---|---|---|
| C-terp V3.5 | PC-DOS | £165 |
| Instant C/16M | PC-DOS | £700 |

### C CROSS COMPILERS

***We supply 2500AD, Avocet, Aztec, Lattice, IAR and Hi-Tech Cross Compilers hosted on MS-DOS and targeted on Z80, 6502, 6801, 68HC11, 6301, 6809, 7811, 8051, 8096, 68000 & 68020.***

## DISK COPYING SERVICE

**We can copy files to and from 600 disk formats including CP/M, CP/M-86, MS- DOS, PC-DOS, APPLE, SIRIUS, BBC, TORCH, APRICOT, HP-150, TRSDOS, ATARI ST, AMSTRAD, MACINTOSH.**

**Our charge is £10.00 + disk + VAT with discounts on small quantities and disks are normally despatched within 24hrs of receipt.**

**For more information call us.**

## ASSEMBLERS

| | | |
|---|---|---|
| 2500AD 80386 ASM v5 | PC-DOS | £230 |
| MS Macro-86 v5.1 | OS/2&PC-DOS | £ 76 |
| MS MASM PDS v6 | OS/2&PC-DOS | £ 63 |
| PharLap 386 DOS Ext SDK | MS-DOS | £350 |
| PharLap 386 ASM/Linkloc | MS-DOS | £905 |
| Optasm | MS-DOS | £110 |
| Turbo Debugger v2 | PC-DOS | £ 89 |
| 2500AD Z80 ASM | CP/M-80 | £ 70 |
| SLR Z80ASM | CP/M-80 | £ 40 |
| SLR Z80ASM-PLUS | CP/M-80 | £140 |
| SLR MAC | CP/M-80 | £ 40 |
| SLR MAC-PLUS | CP/M-80 | £140 |
| SLR 180 (Hitachi) | CP/M-80 | £ 40 |
| SLR 180-PLUS (Hitachi) | CP/M-80 | £140 |

*Not all assemblers are supplied with a linker. Check before ordering.*

## C LIBRARIES & TOOLS

### COMMS LIBRARIES

| | | |
|---|---|---|
| C Asynch Manager V3s'ce | PC-DOS | £135 |
| Essential Comms(MS&TC) | PC-DOS | £180 |
| Greenleaf Comms(source) | PC-DOS | £195 |
| Silvercomm C Asynch | PC-DOS | £140 |

### DATABASE/FILEHANDLING

| | | |
|---|---|---|
| Btrieve for OS/2 | OS/2 | £360 |
| Btrieve for Windows | MS-DOS | £360 |
| Btrieve v5.1 | MS-DOS | £360 |
| XQL | OS/2,PC-DOS | £440 |
| Xtrieve Plus | OS/2,PC-DOS | £360 |
| C/Database T'lchest sce | PC-DOS | £ 39 |
| CBTREE | (source any C) | £135 |
| C-Index/Plus | source any C) | £175 |
| C-ISAM (L,MS) | MS-DOS | £170 |
| C-tree | (source any C) | £215 |
| C-tree Plus | PC-DOS | £360 |
| Essential Btree (s'ce) | PC-DOS | £110 |
| Codebase 4.21 | | £195 |
| CQL s'ce | PC-DOS | £270 |
| Faircom Toolbox Prof v2 | PC-DOS | £680 |
| Pro-C & Workbench v2 | PC-DOS | £585 |
| R-tree | MS-DOS | £160 |
| db-VISTA DBMS Object | PC-DOS | £435 |
| SftFcus Btree&Isam | (sce any C) | £ 80 |

### GRAPHICS

| | | |
|---|---|---|
| Essential Graphics v4 | PC-DOS | £220 |
| GFX Graphics | C-DOS | £ 80 |
| GraphiC v5 (MS) | PC-DOS | £245 |
| graphics MENU/native | PC-DOS | £395 |
| GSS Kernel system | OS2,PC-DOS | £425 |
| GSS Graph Dev T'kit | OS2,PC-DOS | £425 |
| GX Graphics | PC-DOS | £105 |
| HALO v3 (MSC5,TC) | PC-DOS | £215 |
| HALO Professional (C) | PC-DOS | £335 |
| HGraph (MSC) | PC-DOS | £ 80 |
| Ingraf (MSC) | PC-DOS | £190 |
| MetaWINDOW v3.7c | C-DOS | £150 |
| MetaWINDOW/Plus | PC-DOS | £200 |
| TurboWINDOW/C (Turbo) | PC-DOS | £ 92 |
| QuickWINDOW/C (QC) | PC-DOS | £ 92 |

### SCREEN & WINDOWS

| | | |
|---|---|---|
| C-Scape +Look&Feel | OS/2&PC-DOS | £390 |
| Vitamin C for OS/2 | OS/2&DOS | £240 |
| C-Worthy w CW Architect | MS-DOS | £410 |
| Formation Desktop | PC-DOS | £140 |
| Greenleaf Datawindow | C-DOS | £215 |
| Greenleaf Makeform | PC-DOS | £ 75 |
| Mewel | PC-DOS | £250 |
| Panel Plus II (source) | PC-DOS | £260 |
| Vermont Views (MSC,TC) | PC-DOS | £345 |
| Vitamin C (source) | PC-DOS | £270 |

### GENERAL FUNCTIONS

| | | |
|---|---|---|
| C/Utilities T'chest sce | PC-DOS | £ 39 |
| C Tools Plus/6.0 (MS) | PC-DOS | £105 |
| CodeRunner | PC-DOS | £ 95 |
| Entelekon Funct(source) | PC-DOS | £105 |
| Essential C Utils v5 | PC-DOS | £140 |
| G'leaf Fin'cial Mathlib | PC-DOS | £215 |
| G'leaf Super Functions | PC-DOS | £160 |
| Greenleaf Functions s'ce | PC-DOS | £125 |
| Heap Checker | PC-DOS | £ 30 |
| Lattice C Comp Kit+s'ce | PC-DOS | £160 |
| PDK£1 | PC-DOS | £ 70 |
| Power Search | PC-DOS | £ 99 |
| Resident-C (MS,L)(s'ce) | PC-DOS | £135 |
| Silverware C EMM Lib | PC-DOS | £105 |
| The Heap Expander(MS&TC) | PC-DOS | £ 65 |
| Turbo C Tools v2 | PC-DOS | £ 95 |
| WKS Library | PC-DOS | £160 |

### MATHEMATICAL LIBRARIES

| | | |
|---|---|---|
| C Lang Sci Lib v2 | ANY C | £160 |
| Mathpak 87 (L,MS) | MS-DOS | £ 85 |
| Science & Eng Tools | MS-DOS | £ 59 |

### TOOLS & SUNDRIES

| | | |
|---|---|---|
| PC-Lint | OS/2,MS-DOS | £ 85 |
| Clear+ for C | PC-DOS | £140 |
| Databoss (C) | PC-DOS | £320 |
| Lattice Comp Companion | MS-DOS | £ 70 |

### REAL TIME & MULTI-TASKING TOOLS

| | | |
|---|---|---|
| Aspen Multi-C (MSC,TC) | PC-DOS | £ 90 |
| Timeslicer v6 (MS6) | PC-DOS | £230 |
| Over-C (MS) | PC-DOS | £225 |

## LOW PRICES FOR MICROSOFT & BORLAND PRODUCTS

| | |
|---|---|
| **Microsoft Basic PDS 7.1** | **£222.00** |
| **Microsoft C PDS 6.0** | **£222.00** |
| **Microsoft C/Windows SDK** | **£364.00** |
| **Microsoft Cobol 4.0** | **£365.00** |
| **Microsoft Fortran 5.0** | **£205.00** |
| **Microsoft Macro Assembler 6.0** | **£ 63.00** |
| **Microsoft MS-DOS 5 Upgrade** | **£ 60.00** |
| **Microsoft Pascal 4.0** | **£139.00** |
| **Microsoft QuickBASIC 4.5** | **£ 50.00** |
| **Microsoft QuickC 2.51** | **£ 50.00** |
| **Microsoft QuickC & Assembler** | **£ 83.00** |
| **Microsoft QuickPASCAL 1.0** | **£ 50.00** |
| **Microsoft Source Profiler** | **£ 45.00** |
| **Microsoft Visual Basic** | **£ 95.00** |
| **Microsoft Windows 3.0** | **£ 70.00** |
| **Microsoft Windows SDK 3.0** | **£265.00** |
| **Borland C++ 2.0** | **£206.00** |
| **Turbo C++** | **£ 50.00** |
| **Turbo Debugger & Tools 2.0** | **£ 89.00** |
| **Turbo Pascal 6.0** | **£69.00** |
| **Turbo Pascal Professional 6.0** | **£137.00** |
| **Turbo Pascal for Windows** | **£103.00** |
| **ObjectVision 1.0** | **£ 79.00** |
| **Paradox 3.5** | **£340.00** |
| **Paradox Engine 2.0** | **£223.00** |
| **Paradox MultiPack** | **£405.00** |

**WITH FULL TECHNICAL SUPPORT**

## MODULA-2 COMPILERS

*New v3 of Topspeed Modula-2 is now shipping.*

| | | |
|---|---|---|
| Logitech Modula-2 | OS/2 | £400 |
| Topspeed Mod-2 Std | OS/2 | £110 |
| Topspeed Mod-2 Prof | OS/2 | £165 |
| Stony Brook Prof Mod2 | OS/2&DOS | £280 |
| FTL Modula-2 (sml.mem) | MS-DOS | £ 45 |
| FTL Modula-2 (lge.mem) | MS-DOS | £ 55 |
| FTL Modula-2 Developer | MS-DOS | £ 85 |
| Logitech Mod-2/86 Devsys | PC-DOS | £225 |
| Power System w Mod-2 | PC-DOS | £120 |
| Stony Brook QuickMod | PC-DOS | £ 95 |
| Taylor Modula-2 Dev Sys | PC-DOS | £505 |
| Topspeed Modula-2 Std. | PC-DOS | £110 |
| Topspeed Modula-2 Prof | PC-DOS | £165 |
| FTL Modula-2 | Z80/CP/M-80 | £ 45 |
| Modula-2 System | Z80/CP/M-80 | £100 |
| Modula-2 (Mod S/W) | ATARI 520ST | £ 75 |

*Library source is available with some compilers.*

**Novell snaps up DR**

*Following in the footsteps of Borland, Novell has signed a definitive merger agreement with Digital Research (DR) whereby DR will become a wholly-owned subsidiary of Novell. With the merger, Novell is adding DOS, multi-tasking and real time operating systems technology to its UNIX network expertise. Digital Research products, including DR DOS, DR Multiuser DOS and Concurrent DOS, are developed in the the UK in Hungerford, Berks.*

**IBM-Watcom agreement**

*Watcom has announced an agreement with IBM under which Watcom will develop and market OS/2 2.0 versions of 32-bit optimising compilers for C and FORTRAN 77. This adds Watcom to the burgeoning list of developers (including Borland and Zortech) recruited by IBM since the love affair with Microsoft turned sour over OS/2. OS/2 2.0's support for several different execution environments is seen as making it an ideal platform for 32-bit cross-development for a large set of target environments including OS/2, Windows, DOS and embedded systems. The compilers are planned for general availability later this year. More news when we get it.*

**Low-cost Multimedia**

*PictureBook is a new entry-level multimedia package which provides a cost-effective method of producing a presentation that can then be compressed onto a floppy disk. It supports live video and has drivers for laser disk players. All the information is structured and can be accessed either sequentially or selectively using the built-in expert system. PictureBook is priced at £295.00 and is distributed by Digithurst Ltd on 0763 242955.*

**Euro SCOOP**

*The second annual Seminar and Conference in Object Oriented Programming in Europe (SCOOP Europe) will take place at the Church House Conference Centre in London on 28 October - 1 November 1991. For more info or to register, ring Barbara Gavin at the Boston University Conference Office on 071 2592032.*

**Easy OS Switch**

*MultiSys is a menu driven utility that enables a hard disk to switch between more than one operating system in a single partition by exchanging the bootstrap with the operating system's bootstrap and moving other relevant files. MultiSys costs £50 to OS/2 User Group members and £80 to non-members. The OS/2 User Group can be contacted on 0285 655888.*

## Symantec acquires Zortech

The California-based company Symantec has acquired Zortech, the manufacturer of C++ compilers. According to Zortech MD Paul Leathers, this latest in the current rash of software house mergers was entirely friendly, having been planned since early this year.

'Despite our technical expertise, it was hard to compete with the marketing and financial resources of highly capitalised companies such as Borland and Microsoft. Symantec understands our commitment to C++, and can add its expertise to the advancement and marketing of our products' said Leathers. 'The combination of Symantec and our inventiveness will bring about breakthroughs in C++ technology.'

Symantec will take over the distribution of Zortech's current product lines, but they will retain the Zortech name. Symantec is best known for the Q&A database, the Peter Norton range of software, and the 'Think' range of language compilers for the Macintosh. Zortech recently launched V3.0 of its C++ for DOS, Windows and OS/2.

## Environmentally Friendly

Later this month, XTree will be launching a UNIX version of its brilliant MS-DOS file utility program. This takes full advantage of UNIX and allows users to split their system directory structure into several logical *BRANCHES* to allow easy 'housekeeping'. Searching for a file used to be a case of either 'hit' or 'miss'. However XTree has now provided UNIX with a powerful GOTO command that can match a file given only the first few characters of its file-name. Do you remember GREP? I see that you're already poised at your keyboard ready to `man grep`. There's no need. XTree for UNIX allows the user to select a number of files which it then uses to search for a text string. System administrators have a treat in store too. File back-up has been greatly simplified and the COMPACT utility removes directory entries for deleted files, resulting in efficient disk space utilisation and a faster search speed for these directories. XTree can take full advantage of special terminals but will even work with any terminal that supports TERMCAP as long as it has a screen display of 80 columns by 24 rows. XTree for Unix will cost £200 and for every copy that is registered, a real tree will be planted. XTree is distributed in the UK by Bogard Communications on 0753 654333.

## Words, words, words

At a recent press conference given by Microsoft, the company summed up its operating systems strategy in three words, actually one word repeated three times: 'Windows, Windows, Windows!'. The split with IBM seems complete with the disclosure that Microsoft is handing over responsibility for OS/2 version 3.0 to IBM, although development work is continuing at present, in deference to the company's contractual obligations. There was much talk of Windows NT (New Technology) but no demonstration that the product is anything more than vapourware. NT will be implemented as a new 32-bit mode of operation for Windows (along with Standard and Enhanced) and will support pre-emptive multi-tasking and threads. The suggestion that this was, in fact, just another name for OS/2 minus PM was furiously denied by the Microsoft bods.

When NT is shipped, sometime around the middle of next year, users and developers will be faced with quite a dilemma. There will be two similar, but not identical, 32-bit multi-tasking operating systems (Windows NT and OS/2 2.0), with incompatible graphical APIs (Windows and PM), backed by equally powerful organisations (Microsoft and IBM). IBM isn't going to give up OS/2 without a fight and has been busily making friends with Microsoft's competitors such as Borland, Zortech, Watcom and Apple. Meanwhile Microsoft is confident that the success of Windows 3.0 has paved the way for NT to rule the world. If NT really delivers the goods when it is released, and if OS/2 2.0 doesn't live up to the 'better Windows than Windows' hype, the huge installed base of Windows developers and users may well ring the final bell for OS/2.

## More ZEN

Grey Cell Systems will be releasing two new additions to their PC-ZEN networking solution in the middle of September. The PC-ZEN Line Repeater can extend the line length of a network by up to 100 metres. It also increases the number of network nodes that can connected. The PC-ZEN Disk Server enables PCs to share their hard disks and all drives may be accessed from existing applications. It is compatible with Windows and requires only 64 KB for the server and 8 KB for each network node. Each PC-ZEN Line Repeater Unit costs £99.00 and the PC-ZEN Disk Server costs £289.00 providing everything that is needed to set up a complete two-PC network. The network can be expanded at a cost of £79.00 for each extra node. Grey Cell Systems can be contacted on 081 9028998.

# When your system has a weak point . . .

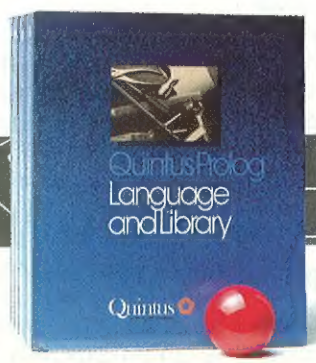
# . . . Quintus Prolog is the strength you're looking for.

Today's practical systems programmers require robust, powerful and productive software tools and more than just one. No one language offers a complete solution in itself. For example, C is a general purpose programming language offering great portability, whilst Prolog gives enormous programmer productivity for searching databases, manipulating strings, reasoning and matching patterns. Each language is powerful when used individually. When used together their potential is unlimited.

Using each language for what it does best is a good solution to the development efficiency problem. With **Quintus Prolog Release Three** complete integration can be easily achieved using C, C + + and other languages.

You can add power and sophistication to applications without adding reams of code. We can prove it.
Call us, on (0923) 211657.

**So avoid weak spots, use the right tools for the job – Quintus and LPA Prolog Products available from A I International Ltd.**




ARTIFICIAL INTELLIGENCE INTERNATIONAL LIMITED
Gresham House, 53 Clarendon Road, Watford, Herts. WD1 1LA
Telephone: 0923 211657 Fax: 0923 247836

CIRCLE NO. 131

**Index support**

*Sequiter Software of Canada has released version 1.04 of CodeBase++ which includes support for Clipper .NTX index files. CodeBase++ is a C++ class library for dBASE database management and now supports the three most popular index formats: .NDX, .MDX and .NTX. CodeBase++ V1.04 comes with a DLL for Windows 3.0 and full source code. The price is £190 from The Software Construction Company (0763 244114).*

**Code Analyser**

*The Performance Analyzer from American Automation is a performance analysis tool for embedded microprocessor systems. It can determine possible subroutine call 'overheads' and then highlight code optimisation by checking the proportion of total CPU time that a given subroutine consumes. Minimum and maximum execution times can also be examined. It costs £1595 and can be obtained from American Automation on 0993 778991.*

**Borland Bumper Pack**

*Borland C++, Turbo Pascal for Windows and Paradox Engine 2.0 all for the bargain price of £499.95. This represents a saving of £250 on the price of the products if purchased separately. The Borland Windows Programming Productivity Pack, as it is affectionately known, is only available from certain participating dealers. For more details contact Borland on 0734 321150.*

**Industrial Objects**

*ONTOS is now shipping version 2.1 of its C++ object database management system for UNIX and OS/2. The new release is described as 'industrial strength' and adds support for IBM's RS/6000 platform. Valbecc, who distributes ONTOS in the UK, says that the product offers significant advantages over relational systems, especially in the areas of telecommunications, networking and multimedia. Prices start at £10,000 for OS/2 and £16,000 for UNIX. Contact Valbecc on 0625 539903.*

**PC-Interface For Windows**

*PC-Interface V4.0 from Locus is an enhanced version of its DOS-UNIX interface program. DOS users can now connect to UNIX and perform printing within a window. Any number of DOS machines may be attached to one or more UNIX hosts. PC-Interface for DOS costs £210 and the UNIX host version costs £260. PC-Interface is distributed in the UK by E92+ on 0634 711700.*

## X-tra Display Screens

When you're running X Windows, don't you find that the screen is too small? Here's something that enables windows to be extended across more than one display screen. QUADRATE is an enhanced X Windows Server and existing applications can run without any modifications. It is compatible with both Open Look and Motif and currently runs on the Sun SPARCstation 2 or IPC. Moving from one display screen to another is merely a matter of moving the mouse in the right direction; the mouse pointer will automatically move across screens. A single CPU license for QUADRATE costs £995.00. For more information contact Quadrate on 0962 884750.


## Higher-Level Graphics

Whitewater has announced the Object-Graphics class library for Turbo Pascal for Windows. It provides a high-level interface to the ObjectWindows library underneath, which will allow portable applications to be written. Graphics objects may be created interactively using the supplied Object-Draw program and there is built-in support for displaying objects like polygons and curves. This greatly reduces development times since these objects handle all of their own screen management. The appearance of graphics objects can be controlled with a number of attributes such as the rendering facility using `Pens`, `Brushes` or `TextPens`. Traditional graphics attributes such as line-style, fill-pattern or colour can, of course, be chosen. The library is already available for Actor and there will also be a version for Borland C++ when the ObjectWindows C++ library becomes available. The ObjectDraw program is supplied with its complete source code. Object-Graphics for Turbo Pascal for Windows costs $195 . The source code is sold separately and costs $195. For more information contact Whitewater on 0101 708 3283800.

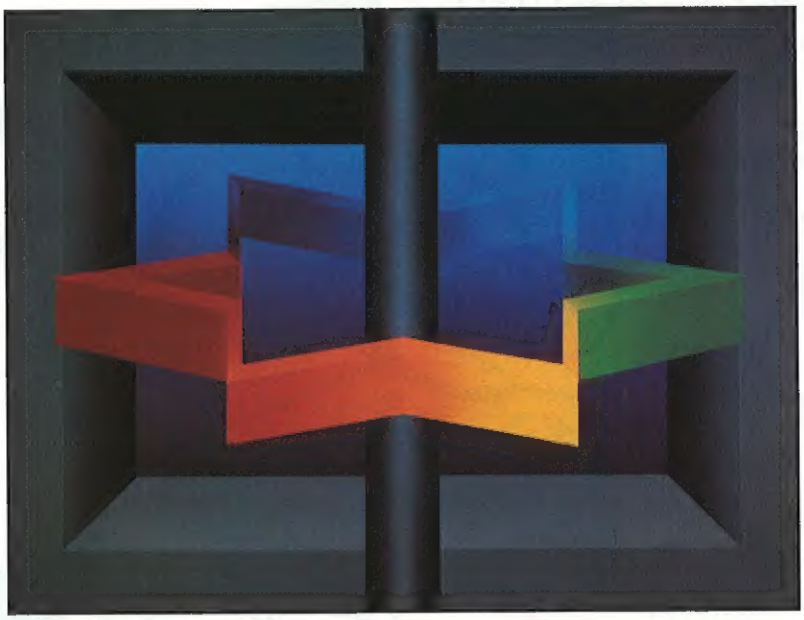
## Novell Developers' Help

Novell hopes that its Professional Developers' Programme (PDP) will encourage developers to create client-based applications, by providing all the development tools and technical support that would normally be required. Software developers will be able to obtain cheaper versions of NetWare and there will also be two tool kits. The Professional Development Series will include tools like network compilers and NetWare APIs. The Software Development Kits will provide the developer with versions of pre-released software. Novell will also be sponsoring a forum on CompuServe called NOVDEV and details of forthcoming events will be sent to PDP members. There will even be a periodical news-letter called *BULLETS* to provide useful technical tips and to review new products. To join the Professional Developers' Programme contact The Enquiry Desk at Novell on 0344 860400.

## Multiuser DOS Tool Kit

Pecan Software has released a development tool kit for the Digital Research Multiuser DOS. It provides a consistent interface to developers who wish to write applications that are portable over a range of hardware platforms/operating systems including OS/2, VAX VMS, SCO UNIX and SUN3. The Pecan Professional Development system costs £300 and includes a compiler and a windowing interface to MultiUser DOS. Contact Pecan Software on 0272 425012 for more information.

# Who put C++ to Work?



glockenspiel
class constructors

### Glockenspiel C++

Glockenspiel C++ leads the field in object-oriented programming. It's the most effective and efficient implementation around. It conforms exactly to the AT&T C++ specification – and they don't come any closer than that! Portable too: it's on more platforms than any other C++ implementation.

### Glockenspiel CommonView®

Glockenspiel CommonView is the leading C++ class library for developing Windows, PM and OSF/Motif applications. Logical, well structured and easy to use, CommonView is 3 to 5 times more productive than the API.

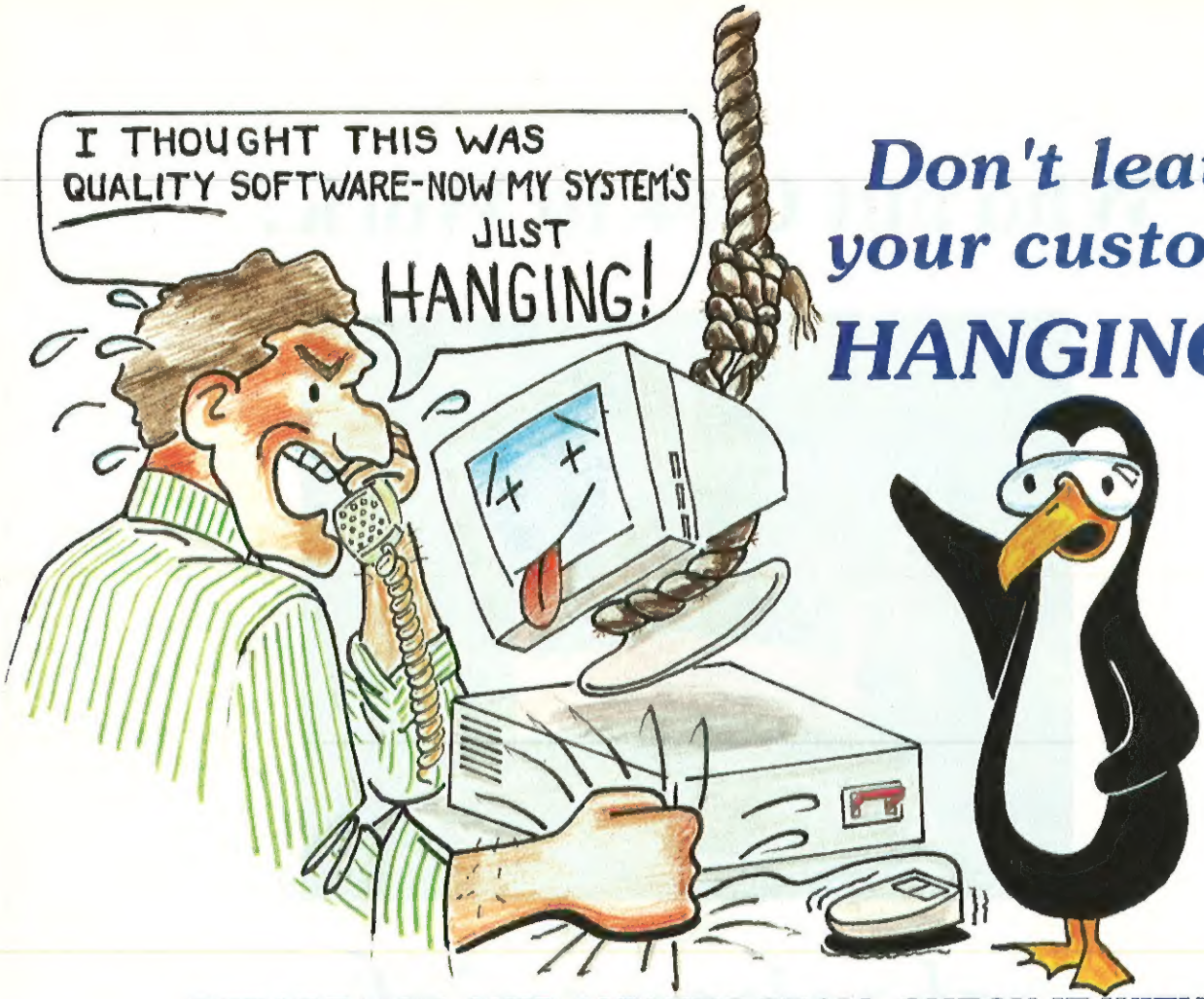
### C++ Training & Consultancy

Glockenspiel C++ training courses reflect our experience as C++ developers. From design to development, *learn from the best!*
Our C++ consultancy services match your needs as you commit more resources to C++ and object-oriented development.

**Further Information**

**GLOCKENSPIEL**, 39 Lower Dominick Street, Dublin 1, Ireland. +353 (1) 733166. Fax +353 (1) 733034

■ **US: IMAGESOFT**, (516) 767-2233. Fax (516) 767-9067. **OASYS**, (617) 862-2002. Fax (617) 863-2633 ■ **UK: QA TRAINING LTD.**, (0285) 655888. Fax (0285) 650537.
■ **Italy: INFERENTIA**, (02) 26680568. Fax (02) 2364258. ■ **Sweden: LINSOFT**, (01) 3124780. Fax (01) 3152429. ■ **Germany, Switzerland, Austria: PSI**, (06021) 492-0. Fax (06021) 492-112
■ **Benelux: RIJNHAAVE**, +31 (71) 218121. Fax +31 (71) 216118




# Don't leave your customers HANGING. . .

## BEFORE YOU SHIP ANY PROGRAM, CHECK IT WITH

COMPUTER LANGUAGE PRODUCT EXCELLENCE AWARD 1990 COLA

# BOUNDS-CHECKER™

WINNER PC MAGAZINE BEST OF 1990

JANUARY 15, 1991 PAGE 168

### FINDS OUT-OF-BOUNDS MEMORY ACCESSES *AUTOMATICALLY*

Your program may have 10,000 to a million lines of code. It may occasionally hang mysteriously or it may appear to run flawlessly every time. But under DOS, how can you ever be sure that your program is not corrupting memory it does not own? The only way to be 100% sure is to BOUNDS-CHECK before you ship.

**To use BOUNDS-CHECKER** you build your program with debugging information (we support most compilers including Microsoft, Borland & JPI). Then you just type <BC file-name>. BOUNDS-CHECKER sets up the 386™ /i486 for protection and lets your program fly. If your program accesses memory it does not own or overwrites its own code, BOUNDS-CHECKER pops up displaying the offending SOURCE-LINE or instruction.

**Programming under DOS is a gamble,** so why not stack the odds in your favor--CALL TODAY.

**(603) 888-2386**

### OTHER FINE NU-MEGA PRODUCTS. . .

**The ultimate systems debugger.**

**Debug:**
- Interrupt routines
- Device drivers
- T&SRs
- ROMS
- Applications
- Overlays

Soft-ICE™

**Features:**
- Break out of a hung program
- Real time Break-Points
- Back-Trace history
- Works with other debuggers

If you are debugging an application, Soft-ICE is seamlessly integrated with BOUNDS-CHECKER so you can easily go back and forth between BOUNDS-CHECKing and debugging: a combination many programming professionals can't live without.

1988 BYTE AWARD OF DISTINCTION

**Run CODEVIEW for Windows on a single monitor**
**CV/1.....$129**
• Runs in a window • No annoying flash •
• Runs on any display that supports windows •

Call by 4:00 PM EST TODAY and ask us to EXPRESS you an info packet. In most cases you will receive it by 10:30 AM tomorrow. (USA only)

All Nu-Mega products require a 386, 386SX or 486. MS-DOS and Codeview are trademarks of Microsoft Corp. 386 is a registered trademark of Intel Corp. Nu-Mega, BOUNDS-CHECKER, Soft-ICE and CV/1 are trademarks of Nu-Mega Technologies, Inc.

Nu-Mega™ TECHNOLOGIES INC

P.O. Box 7780 • Nashua, NH • 03060-7780 U.S.A.
(603) 888-2386 • Fax (603) 888-2465

BOUNDS-CHECKER ..... $249
Soft-ICE ..... $386
**Special Offer**
**Buy BC & S-ICE .....Save $100**

30 Day Money Back Guarantee

CIRCLE NO. 118

## Goldilocks

I'm still puzzled by the analogy, but Insignia Solutions has titled its new SoftPC family 'Goldilocks and the three bears'. SoftPC provides a method of integrating the Macintosh into a PC-based office environment. Goldilocks (SoftNode V1.0) costs $175 and allows the Mac to be connected to a Novell network by linking into its Macintosh Ethernet Driver software. The base system (baby bear) is priced at $199 and requires a Mac with at least 2Mb of memory and a further 2MB of disk space. It is capable of emulating a text-based PC application, although it has a limited graphics capability; mother bear (Universal SoftPC V2.5) can be run on any Mac and provides support for Novell but it still has only a limited graphics capability and costs $399. Father bear (SoftAT V2.5) can emulate EGA but requires a 68040 based Mac and costs $542. I wonder who's been eating my porridge? For this, and any other questions, contact Principal Distribution on 081 6777631 or 0392 434071.

## Modula-2 Conference

If you're quick, there's still time to register for the Second International Modula-2 Conference, to be held from the 11th-13th of September at Loughborough University. The keynote speaker is Prof Gustav Pomberger of the University of Linz, Austria, who is tipped by the software engineering cognoscenti as Niklaus Wirth's successor. Other speakers include Greg Nelson from DEC Palo Alto, a co-designer of Modula-3, and Juerg Guttknecht of ETH, Zurich who helped Wirth with Oberon. Professor W himself will also be present. Registration costs £150, details on 0509 222174.

## Applications Browser

Applications Browser from the Hypersoft Corporation is a COBOL reverse-engineering tool which enables developers to maintain an up-to-date copy of documentation. Documentation on existing code is generated automatically when it reads the source files, building a set of *CodeBase* files. The CodeBase can then be examined at any time. Source files are first analysed producing a report which points out anything that's unusual about the flow of the program or occurrences of undefined data references. This information may be displayed either graphically or textually. There are also CALL, PERFORM and GOTO charts which enable the program structure and code to be examined at a glance. Applications Browser for the PC costs £2678 with an annual support charge of £365. It is available for a number of other platforms including VAXs under VMS and is distributed in Europe by Scientific Computers on 0444 235101.

## Comms Workstation

CentrePoint Technology has released a new terminal emulation product called pcVision. It provides an accurate emulation of almost any terminal on a PC by supporting 132 Column Displays, downloadable fonts (to match the fonts of the emulated terminal) and it emulates terminal keyboards precisely. pcVision supports numerous file transfer protocols including Kermit and XMODEM. File transfer or printing occurs in the background enabling the PC to run other applications. There is a command language that may be used to automate more complex procedures and the built-in Modem Dialer contains a database and will automatically configure the modem. In addition, pcVision provides an API to allow developers to use the communications software in their own applications. The pcVision/4 product costs £225, and the API costs an additional £30. Contact CentrePoint Technology on 0733 323010 for more information.

## WASP?

Sounds a lot like a heavy-metal rock band, but in fact, WASP actually stands for Windows Association of Shareware People. Membership is free. It is hoped that WASP will organise comprehensive technical support on Windows programming. Shareware authors of Windows software have been asked to contact WASP with details of their applications and copies of their distribution disks. WASP can be reached on 0373 865263. Meanwhile, I'll get out my *Gibson*.

### Total Security

*The Disk Encryption Unit from Tulip computers protects the data on a hard disk, even if it is stolen. This is achieved using an IDE controller that encrypts data. Information on the disk can only be accessed by using a valid password on entering the system. The Disk Encryption Unit costs £325. For more details contact Tulip computers on 0293 562323.0*

### Mac Comms

*Paul Smith, a regular .EXE contributor, would like to spread the word on his CommsTalk V2.0 utility for Hypercard. CommsTalk provides a communications interface on the Macintosh that allows Hypercard developers to build such applications as GUI front-end for mainframes and multimedia. Paul may be contacted on 0491 574295 for more information.*

### Parallel C

*3L is shipping the latest version of its parallel C compiler (V2.2) for the transputer. The new release can now generate code for the complete transputer family and a Message Broadcasting facility has been added. 3L's Parallel C V2.2 costs £595.00. For more information phone 3L on 0506 415959.*

### Paradox Compiler

*PALCom is a compiler that can produce executables from database applications written using the Paradox Applications Language (PAL). As the application is compiled instead of interpreted, there should be a marked improvement in performance. PALCom costs £319 and is distributed in the UK by The Software Construction Company on 0763 244114.*

### Evaluating X

*Tektronix are offering a validation programme to provide developers of X applications with a method of checking whether their software is compatible with TekXpress terminals. Under the scheme, software is submitted for evaluated. Tektronix may be contacted on 0628 486000.*

### C++ Reference

*A little misleading perhaps? Actually its a rather neat little booklet packed with 16 pages of vital information on C++. The reference covers C++ by examples and there are also statement formats and classes. The C++ Reference costs $4.50 and is available from Specialised Systems Consultants on 0101 206 5273385.*

## VB Tools

The Control Development Kit (CDK) and Windows Help Compiler are now available for Microsoft's Visual Basic. The CDK allows the development of custom controls for the VB Toolbox. The Help Compiler enables VB programmers to add on-line help to their applications using the Windows help engine. Both products require Windows 3.0 and the VB programming system. To use the CDK you also need the Windows SDK and Microsoft C V6.0. The Help Compiler requires a word processor that can generate Rich Text Formats (RTFs), such as Word for Windows. Both products are priced at £39. Microsoft is on 0734 500741.

# Letters

*We welcome short letters on any subject that is relevant to software development. Please write to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. Unless your letter is marked 'Not for Publication', it will be considered for inclusion on this page.*

## Horse Feathers

Sir,

I found the cover of your August issue in extremely bad taste.

My Grandfather's horse, Hercules, was commandeered by the Ministry of Defence during the First World War and unfortunately died in a gas attack.

Your cover caused me much distress and I will be cancelling my subscription forthwith.

*Mr ST Eptoe*
*March*
*Cambridgeshire*

## But seriously...

Sir,

I could not believe the tastelessness of August's cover. Quite apart from the electronics catalogue level of intelligence which it assumes. Do you realise:

1. The soldier is wearing a mask to protect him from gas attacks; attacks which sear the lungs and destroy the brain. 2. Statistically, the soldier was dead in four years (I presume that the picture was taken between 1914 and 1918). 3. The war in which the soldier is participating was the bloodiest most wasteful of human life, ever.

This is not a suitable subject for a technical magazine cover, particularly when it is supposed to be light-hearted. Will you be showing, eg abused children (to illustrate market penetration!) or AIDS victims (to illustrate wasting-away of old programming products)? How about Christ on the cross, with OS/2 painted on his lacerated chest? Very bloody amusing, eh?

I hope you will stick to humour next month.

*Michael Mounteney*
*Peterborough*

## Poxy Objection

Sir,

With reference to our press release for BRIEForC++ which appeared on page 4 of the August edition of .EXE.

I object most strongly to you using the word 'poxy' to express your opinion of the name chosen by Solution Systems for their C++ add-on for BRIEF. If you wish to use derogative *(sic Ed)* adjectives to describe your opinions you should at least use a 'proper' word and not slang.

*Roger Lee*
*Solution Systems UK Ltd*
*Hertfordshire*

## Mono problems

Sir,

I recently shelled out for a monochrome adapter and monitor (sounds daft I know, but I couldn't find any on the local scrap heap) in order that I might debug my Windows applications with CodeView. I immediately came upon a problem in that the horizontal synch of the mono monitor wouldn't lock. CodeView Version 2.x was OK but version 3.0 and for Windows wouldn't work.

It transpired that the problem was caused by a write of 00H to indexed register 23H of I/O port 3B4H. This port on the mono card is not uniquely decoded and wraps onto indexed port 3, the End horizontal blank register, and should be written with 0FH.

Upon further investigation it became apparent that CodeView issues an Int 10 function 5, Select active page. This results in the VGA Video BIOS issuing the I/O since it is not 'aware' that another card is in use. My interim solution has been to patch the RAM copy of the VGA BIOS so that location C000:83E reads MOV AX, 230FH for the Orchid BIOS. Note that this problem will only occur on machines such as mine with a Tseng Labs ET3000 VGA chip.

If any other developers out there have come across this they might appreciate these findings.

*Roger Campbell*
*Springborn Data Ltd*
*Surrey*

## DMA appeal

Sir,

I have recently been looking into programming the DMA controller on a 80486 PC (AST). I find that the DMA controller on a 486 is not the same as on the 386 and below, and the PS/2. What I also find is that most books which are about programming the PC hardware have a single reference to DMA as 'a means of inputting or outputting data from memory without using the CPU. New paragraph, new subject' or some other equally insignificant statement.

What I would like to find is a book that will tell me how to use the DMA controller effectively, and indeed other bits of PC hardware, particularly timers and the PIC. I am actually interested in controlling a 80186 based fax card in a 80486 PC.

Could you please advise me? Have there been any articles in .EXE in the darkest past? I have been reading .EXE for several years now, and cannot remember any. I must also say that I enjoy reading the magazine. These days, it seems to be the only thing that comes through the letter-box which I actually want!

*N R Mason*
*INFOGEN*
*Avon*

*We don't know of any .EXE article or book which addresses this problem. Are there any better-informed readers out there?*

## Critical Locke

Sir,

I question both the utility and advisability of the techniques described in the article 'C++ Member Functions in Assembly Language' *(.EXE Magazine, August '91)*.

The Zortech manual does indeed 'suggest that you avoid writing assembly with C++ linkage'. The C++ compiler provides inline functions, and these can be used to create a C++ member function which calls an assembler function with C (or, indeed, Pascal) linkage without significant overhead.

If you could create your own computer for developing software what would you build?

You'd need a system to increase your productivity, one that helped you create applications in half the time it takes you now – and complete projects ahead of schedule! And, with only a little training, you'd need to be able to create customised applications that met all your user's needs, with draft layouts to show them produced in just a few days.

You'd want a true multitasking system with virtual memory. Because that would give you the kind of computer that will let you compile a program in one window while debugging in another, so if your new application should crash your whole system won't die.

You'd give it plenty of power and speed to handle even the most complex projects, complete with UNIX and the Motorola 68040 processor.

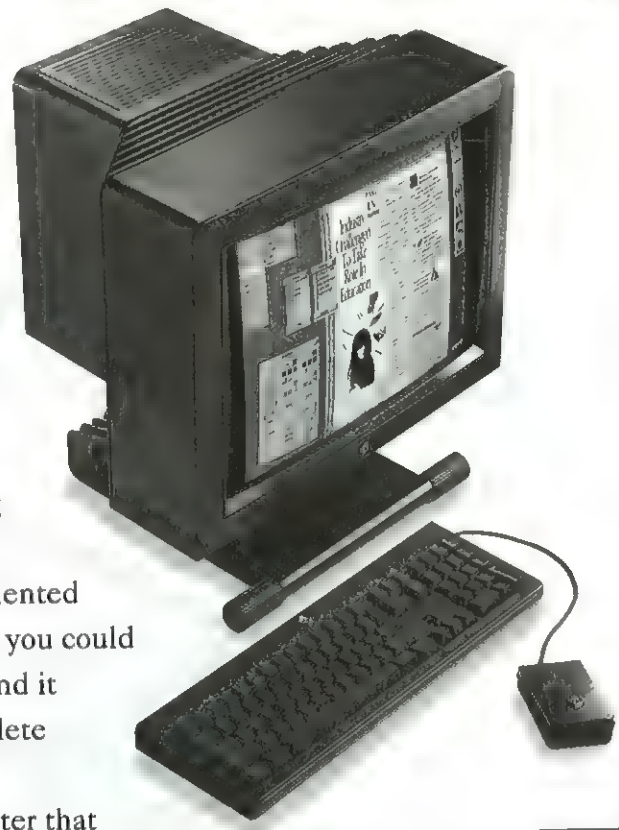
With your users in mind, you'd want an Interface Builder so you could create an outstanding user interface – quickly and easily.

And you'd want an object oriented programming environment so that you could reuse your software components and it would help if it came with a complete object library.

Fortunately, there's a computer that already offers this, the NeXT computer. The tools provided by the NeXTstep operating environment make it easy to develop advanced, user-oriented programs.

So take the NeXTstep and call 0800 898488 for more information, or fill out the coupon.

# THE ULTIMATE DEVELOPMENT MACHINE.



CIRCLE NO. 110

The Professional's WorkStation

I'd like to receive more information on the NeXT Developer Program. ☐

I'd like to attend a development seminar ☐

Please complete and return to: NeXT, 286 Bath Road, West Drayton, Middlesex UB7 0DQ.

Name:

Company.

Address:

County:

Telephone: Post code:

EXE 1

The only occasion where there would be any loss of efficiency is when the member function is virtual, **and** when the class to which an object belongs cannot be ascertained at compile time.

A situation where a virtual function is in such a time-critical area that the overhead of an extra function call cannot be tolerated is extremely rare. In any case, the first thing to consider in such a situation is to make the function non-virtual (which would improve the efficiency by itself).

I would certainly be very reluctant to take the risk of coding C++ linkage functions in assembler when the compiler vendors all have different name mangling and calling conventions, and even the individual vendors warn that the conventions may be changed in future versions of the compiler. Users of the original Lattice C will know what a pain this is!

Also, the rather incomplete description of vtables (described as 'scary' in the article) is not good enough to enable calling virtual functions from assembler - I believe the first entry in the vtable has something to do with multiple inheritance, which is not even mentioned in the article. I also suspect that the layout of the vtable will change depending on whether the '-g' flag is used (debugging information is added to the end of the vtable, I think).

While I am slagging off the August issue, I was not impressed with the 'Upper, Middle or Working Class' article. It used up more space than was necessary for a series of product announcements, and fell far short of being a comparative review. It has not helped me make a buying decision at all.

Don't take these criticisms too much to heart, though - most magazines do not have such a high standard to live up to!

*Nikki Locke*
*Shropshire*

*I accept that Laine Stump's C++ article described techniques that operate at a lower level than most readers would ever need to use; and that the information presented was incomplete. This second point was my fault: I was not prepared to give Laine the extra space he needed to go deeper into the subject.*

*The rationale for publishing the article in the form it appeared was as follows. Those who had no immediate practical use for the information might nonetheless gain useful insight into the workings of C++ compilers. For those who do want to call virtual functions from assembler, the article offered a reasonable starting point from which to extend their knowledge.*

*The 'Upper, Middle or Working Class' article was intended only as a general introduction. We tried to cover as many libraries as possible, with the intention of alerting our readers to the breadth of choice. More detailed articles on individual libraries will follow, starting with the piece on Zinc printed elsewhere in this issue - Ed.*

## What's he on?

Sir,

Whilst reading Borland's C++ primer I have just had what can only be described as an 'out of body experience'. I wonder if any of your other readers have had similar experiences?

*Richard Samworth*
*Software Components Limited*
*Beds*

*Letters submitted to this page may be edited. The writer of the best letter of the month, as judged by the Editor, will be rewarded by a T-shirt or similar-valued .EXE trinket. The best letter is the one printed first.*

# Full Duplex Communications Link in C++

*Use the full bandwidth of your data connections - in BOTH directions simultaneously - with Laine Stump's character stream based, full duplex, reliable Connection class.*

While most communications libraries support reliable file transfers in one direction at a time, and can send characters across the line in both directions (with no error checking), they are seriously lacking when it comes to setting up a reliable, character oriented connection, or sending in both directions simultaneously.

## What about X/Y/Zmodem?

Every library has support for Xmodem and Ymodem, and at least one (Solid Link) even does Zmodem. These error checking protocols are standardised, and they're useful if you want to send files, but that's where their functionality ends. A major problem is that all these protocols are half-duplex - information can only go in one direction. Although their specifications have true 'packets' running one way, the reverse line only carries simply ACK and NAK (Negative AcKnowledgement) signals - there is no packet type information to allow data packets to co-exist with ACK and NAK packets.

Most comms libraries implement these protocols for transferring files, not generic information. So if you're going to send some data, it's a three step process: write the data to a file, send the file, and read the data from a file at the other end. And if data is flowing in both directions, you'll have to develop some type of 'over, over and out' signalling to allow switching the direction of communications. That's a big waste if the two machines are having a two way conversation ('Send me yesterday's sales totals' 'Ok, ...' 'What about last week?' 'Not so good ...').

A good example of a system which cannot be easily and efficiently implemented using these commonly available 'plug and play' protocols is a networking module for a bulletin board system. The idea here is that, at some pre-arranged time, two BBSs will contact each other, exchanging all messages that have been entered since the last net connection. Each BBS should have the ability during the net session to request which messages it wants to receive ('send me the Politics, but leave out the Stock Tips'). Both sides of the connection can make requests (commands) and send messages (data).

With systems like this, implementations using X/Y/Zmodem are extremely wasteful. Besides sending data in packets of at least 128 bytes (filling in any unused bytes with 0 or garbage as Zmodem packets are of variable length), the inherent half-duplexity of X/Y/Zmodem is a real waste of datalink bandwidth, and, therefore, of telephone time (and, consequently, **money**!). Just think of it, while node A is sending a packet of data to node B, node B just sits waiting for the end of the packet, when it will send an acknowledgement (ACK) that the packet has arrived safely. Then A will wait while B sends a packet making another request etc (see Figure 1).

**HALF DUPLEX**

| *Machine 1* | *Machine 2* |
|---|---|
| req A | (idle) |
| (idle) | ACK req A |
| (idle) | send A |
| ACK send A | (idle) |
| (idle) | req B |
| ACK req B | (idle) |
| send B | (idle) |
| (idle) | ACK send B |

**FULL DUPLEX**

| *Machine 1* | *Machine 2* |
|---|---|
| req A | req B |
| ACK req B | ACK req A |
| send B | send A |
| ACK send A | ACK send B |

*Figure 1 - X/Ymodem (half duplex) vs Connection (full duplex)*

## Design Goals

In my design of a protocol to eliminate this waste, I had three desires: to communicate reliably, on a character rather than packet basis, and as quickly as possible. To aid fast communications, I made two major design decisions:

- 1) the protocol must be full duplex, allowing both data and ACK/NAK signals to flow in both directions simultaneously.
- 2) packets should not have a fixed length; the data of a packet starts when a Packet Start sequence is received, and ends with a Packet End sequence.

I also had a third design decision which I had to drop due to time constraints.

- 3) a new packet can be sent without waiting for the ACK of the previous packet. This is called 'packet windowing' and will come in the future, but for now 1 and 2 are enough.

## Tools

The program implementing the protocol discussed in this article was compiled with Zortech C++ version 3.0. I hate duplicating previous effort, so when I needed a linked list class I used the `zGSList` class from the Zortech C++ Tools library. `zGSList`

# Development speed = Oregon C++

Oregon C++ is, quite simply, the fastest C++ Development System software you can buy. Consisting of a true optimising compiler (not just a C translator), a source level debugger and libraries, it sets speed records in every area. Here's how.

**The optimising compiler means optimum speed**

Because the optimising compiler directly generates compact object code it eliminates the translating step. The result? Faster compilation, direct debugging and faster program development. And you end up with smaller and faster applications. What's more the compiler is switch selectable for C++, ANSI C, or K&R C.

**You don't waste time worrying about compatibility**

Oregon C++ conforms to ANSI standards for the C++ language and supports inter-language calls to and from C, Pascal, Modula-2 and Fortran. So you get fast, easy, access to existing code modules, without wasting time rewriting or re-debugging. And Oregon C++ is totally compatible with all existing C libraries.

**Fast debugging is done in the same language in which the application is written**

The Oregon Debugger - ODB - debugs C++, ANSI C and K&R C in the original application language, which means you get more reliable code in less time. And because the ODB understands multiple inheritance, it can quickly display the class hierarchy.

**Oregon C++ is fast and easy to use**

With a choice of command line or mouse-driven window user interfaces, Oregon C++ is easy to use. In window mode you even get separate windows for the application and debugger I/O, and every window is fully scrolling so you can find the information you want in no time at all.

**Fast information...**

For an instant response to your request for information, call us today. Or clip the coupon for your free technical data.

**Instrumatic**

The Pan-European Technology Group
**Instrumatic UK Ltd.**
First Ave., Globe Park, Marlow, Bucks. SL7 1YA
Tel 0628 476741 Fax. 0628 474440
Tlx 847042 IMATIC G

**For more information fill in the coupon and post today!**

Name

Position

Company

Postcode Telephone

.EXE 9/91

CIRCLE NO.114

## C-scape® 3.2

The latest release of C-scape 3 — the best selling character/graphics based screen interface C library in the UK. Runtime royalty free. Includes Look and Feel Screen Designer™ and Source Code. DOS, OS/2 **£370**
Also available for X Windows, UNIX, VMS and most DOS Extenders.

## Heron Software C-scape Extension Library

**New version 2**
Enhanced set of functions for C-scape 3 Price
Includes Source Code..... DOS **£150**
Unix **£250**

## Phar Lap 386|DOS Extender™

Beat the 640K DOS limit with a flat 32-bit address space on 386 processors. Voted best of 1989. Supports a wide range of 32-bit languages. .....SDK **£315**

## Phar Lap 286| DOS Extender™

C beyond 640K with your Microsoft C compiler on any 80286, 386 or 486 platform....SDK **£315**

## db_VISTA™ III — Release 3.2

Complete family of C library dbms functions and database management utilities for DOS, Windows, OS/2, Macintosh, UNIX, QNX, VMS. Source code available — runtime royalty free.
Single-user DOS, OS/2 or Windows DBMS object libraries..... **£435**
Ask for details of other options and multi-user pricing.

## MagnaCharter II

Enhanced version of the much acclaimed MagnaCharter charting package for Windows 3, 99x99 cells with wide range of printer support. Complete set of symbols — or add your own..... **£160**

## MagnaCharter

Still available, MagnaCharter for DOS (includes runtime Windows 2.)..... **£118**

## PERSONAL ACCOUNTANT

The only complete personal finance package — for DOS **£149.95** (including VAT).
**NEW — Tax Reckoner for UK tax payers — for DOS £119.95 (including VAT).**

**All Systemstar products are distributed and supported by Systemstar. Prices are exclusive of VAT (except Personal Accountant) and subject to change. Orders and enquiries to:**

**SYSTEMSTAR LIMITED**
1-3 Parliament Square, Hertford, SG14 1EX
Telephone: (0992) 500919 Facsimile: (0992) 554261

CIRCLE NO. 115

has three functions that concern us: `insert()`, `append()`, and `get()`. `insert()` puts a new entry at the head of the list, `append()` puts an entry at the end of the list, and `get()` takes an entry from the head of the list (or returns 0 if the list is empty). You could easily implement these yourself, or use a different linked list implementation, if you don't have the Zortech C++ Tools.

To make the implementation easier to understand, I wanted data receiving and sending to be handled by separate processes. Rather than cooking up some cheap, rickety task scheduler myself, I used the Schedule library included with the Solid Link Communications Library. As well as the necessary functions to start and stop processes, it also has functions for mutual exclusion semaphores, important when two processes are simultaneously making calls to a non-reëntrant run-time library and DOS. All these functions are preceded with 'sch', and their names are self explanatory (eg `schResume()`). If you're using OS/2, you can replace these calls with the appropriate thread management calls. Otherwise, if you don't have Solid Link, you'll have to find or implement another multi-tasking library, or seriously rework the whole implementation to work as a single process.

Finally, I used the basic communication functions from the Solid Link library for the 'physical' link. These functions are all preceded by 'mdm', and also have descriptive names (`mdmRead()`, `mdmWrite()`, `mdmInstall()`). They are only used in the final derived class `SerConnection`, and can be replaced with any equivalent code to perform the same functions.

## *X/Y/Zmodem are half-duplex protocols*

## Overview

Class `Connection` is implemented as two virtual base classes, `Protocol` and `Connection`, which will be inherited by a user defined class containing the details of the physical connection. The same protocol can be used with vastly differing hardware; all that is necessary is a new derived class with an initialisation function (the constructor), a `RecvChar()` function, and a `SendChar()` function, which are declared as virtual in the base classes. As separate processes are calling `RecvChar()` and `SendChar()`, there is no need for status functions; if there is no character available, the process just waits until one comes along.

## Packet

`Packet` is a class which constructs and breaks down packets of information. All communications on the `Connection` are sent in `Packets`, including ACKs and NAKs. Each `Packet` has a `Type` (DATA, ACK, NAK, RESET) and a `Seq` (sequence number). `Type` is necessary to allow mixing data and ACK packets in the connection. `Seq` is included in the definition of `Packet` for two reasons: first, to aid in catching 'false ACKs' (when an ACK for a previous packet is mistaken as an ACK for the just-sent packet), and second to allow for a future upgrade to a windowing protocol, which would allow multiple packets to have outstanding ACKs at the same time.

The only member of Packet large enough to warrant a non-inline function is `CkSum()`. It generates some sort of check-sum of the packets fields which will be sent across the `Connection` as well as computed at the other side, to verify the data. My example `CkSum()` function is quite simple; you may want to replace it with a more elaborate (and reliable) CRC computation.

```
// connect.h
//  all necessary class definitions for
//  a full duplex, variable length packet
//  character stream communications protocol
//  - Laine Stump, July 1991, No Rights
Reserved -

#include <schedule.h> // from SolidLink Com
Library
#include <modem.h>
#include <gslist.hpp> // from Zortech C++
Tools
class Packet;
declare(zGSList, Packet);

// shorter names for securing exclusive use
of DOS
inline void schWaitLibSemaphore()
  { schWaitSemaphore(schCLibSemaphore); }
inline void schFreeLibSemaphore()
  { schFreeSemaphore(schCLibSemaphore); }

const TIMEOUT = 100;// 10ths to wait before
time-out
const PSIZE = 2048; // max. packet size
const CAN = 0x18, END = 32;

enum PacketTag { DATA = 33, ACK = 34,
                 NAK = 35, RESET = 36 };
class Packet
  {  // a Packet of Data in Memory
private:
  PacketTag Type; // kind of packet
  int Seq;        // which packet
  int Size;       // max entries
  int Valid;      // # bytes of data
  int Out;        // next to Get
  char *Data;
public:
  Packet(PacketTag t, int s = 2048, int seq
= 0)
    { Type = t; Seq = seq; Size = s;
      Valid = Out = 0; Data = new char[s]; }
  ~Packet()                    { delete
Data; }
  void SetSeq(int s)           { Seq = s; }
  void SetType(PacketTag t)    { T,pe   t; }
  void Empty()       { Valid = Out = 0; }
  int IsEmpty()      { return (Valid == 0); }
  int IsFull()       { return (Valid ==
Size); }
  int IsExhausted() { return (Out ==
Valid); }
  PacketTag TypeOf(){ return Type; }
  int SeqOf()        { return Seq; }
  int CkSum();
  // Check for Empty or Exhausted before
Put & Get
  void Put(char c)   { Data[Valid++] = c; }
  char Get()         { return Data[Out++]; }
  void Again()       { Out = 0; }
  ;;  // class Packet
/*
  As sent, a Packet looks like:

    CAN [type] [seq] [data] CAN END [cksum]

  [type]  - DATA, ACK, NACK, RESET
  [seq]   - packet sequence number.
  [data]  - any data (DATA packets only)
  [cksum] - 16 bit additive checksum of
[type][seq]
            [data], plus the length of the
packet.

  all bytes are sent as normal, unless a
  CAN is  encountered  ANYWHERE (including
[seq],
  [data], and [cksum]), in which case it is
sent as
  (CAN CAN+0x40).

  This is all handled by XmitPacket
  and GrabPacket (in Protocol)
*/

class Protocol
  {  // class which handles all protocol de-
tails
private:
  // packet lists used by Sender & Receiver
  zGSList(Packet) ToSend, Recvd;
  int SendSeq, RecvSeq;
  Packet *Sending, *WaitingACK, *Recving;
  int Done;  // set to 2 to terminate
  void SendCharCAN(char c);// send, adding
CAN
  int RecvCharCAN();    // recv, interpret
CAN
  void XmitPacket();    // send one packet
  void GrabPacket();    // recv one packet
protected:
  tPID SendPID, RecvPID;
  // defined in final class xxxConnection
  virtual void SendChar(char c) = 0;
  virtual char RecvChar() = 0;
public:
  // called by processes spawned in xxxCon-
nection
  void PacketSender();  // Packet sending
process
  void PacketRecver();  // Packet recving
process
  Protocol();
  ~Protocol();
  // schedule a Packet to be sent
  void SendPacket(Packet *ToGo);
  // get the next valid Data Packet (WAITS!)
  Packet *RecvPacket();
  // return next Packet if ready, else 0
  Packet *TryRecvPacket();
  int IsIdle()  { return !WaitingACK &&
!Sending; }
  void Suspend(){ schSuspend(RecvPID); }
  void Resume() { schResume(RecvPID); }
  };  // class Protocol

class Connection : public Protocol
  {  // a virtual base class for all connec-
tions
protected:
  Packet *Going;  // chars going to Protocol
  Packet *Coming; // chars recved from
Protocol
  virtual void SendChar(char c) = 0;
  virtual char RecvChar() = 0;
public:
  Connection() : Protocol()
    {  Coming = new Packet(DATA, PSIZE);
       Going  = new Packet(DATA, PSIZE); }
  ~Connection()  { delete Going; delete Com-
ing; }
  int  Ready();
  char Get();
  void Put(char c);
  void Flush();
  int IsIdle()
   {return Going->IsEmpty() && Proto-
col::IsIdle();}
  };  // class Connection
// end of connect.h
```

*Figure 2 - CONNECT.H*

RAM PAGEFRAME IPX MEMORY BIOS CACHE VGA MOUSE 386 BLUEMAX 486 HIGH DOS EGA INITIALIZE

# Only the most intelligent memory managers can pass this test.

## There's no question.

If you want top performance from your 386 system, you need maximum memory management. And that takes intelligence. But whose intelligence would you rather use - yours or your memory manager's?

Here's a little quiz to help you make the smart choice.

**1 True or False: All memory managers are alike.**

False. Most memory managers free up space for applications by moving TSRs and device drivers from conventional memory into high DOS memory. But they vary widely in how effectively they do it. Others require a lot of guesswork, and a lot of time. And you still won't get top performance.

MAX, on the other hand, uses its intelligence to calculate automatically the thousands of possible ways these programs can be arranged in high DOS, and finds the best possible fit. First time, every time. Guaranteed.

**2 Why do other memory managers leave some programs in conventional memory when there's still room for them in high DOS ?**

Many resident programs need much more space to load than they need to run. FlexFrame, a MAX exclusive, "borrows" up to 64K of high DOS memory for loading, so it can pack more in. That frees even more memory for applications.

**3 True or False: Using TSRs in Windows is a great way to crash your system.**

With other memory managers, that's definitely true. But not with MAX. Thanks to another MAX exclusive called TSR *instancing*, you can use nearly any pop-up utility as many times as you want in Windows.

**4 With DOS 5.0, there's no need for a memory manager - right or wrong?**

Well, that depends. DOS 5 does free up some memory with its smaller program size. And it does let you place programs into high DOS manually.

But for optimal memory management and guaranteed top performance, what you really need is MAX. It's the powerful, auto-mated, full-service program that specializes in all facets of managing your memory.

So while DOS 5 is good, DOS 5 with MAX is outstanding.

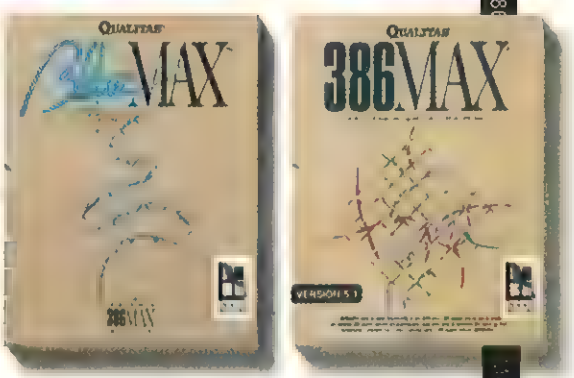
## The answer is easy

Just choose the MAX that's right for you. There's 386MAX® for any IBM compatible 386 or 486 based system at just £79*. And BlueMAX™ at £99*, the only memory manager that gives you up to 250% more high DOS memory on 386 and 486 IBM PS/2s.

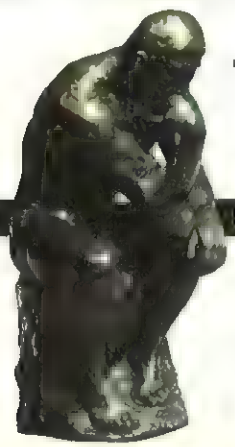
Order direct and we'll have MAX on your desk the next business morning. We'll even include a free copy of ASQ™, the intelligent memory analyzer.

To get maximum performance from your system, graduate to the Intelligent Memory Managers. Call READMAR SYSTEMS

**(071) 625 5255**




# The Intelligent Memory Managers™



RAM MEMORY BIOS CACHE VGA MOUSE 386 BLUEMAX 486 HIGH DOS EGA INITIALIZE

 **System Requirements: Any 386 or 486 PC or PS/2, min.256KB of extended memory, DOS 3.0 or higher, high density floppy disk or hard disk drive.**

QUALITAS®

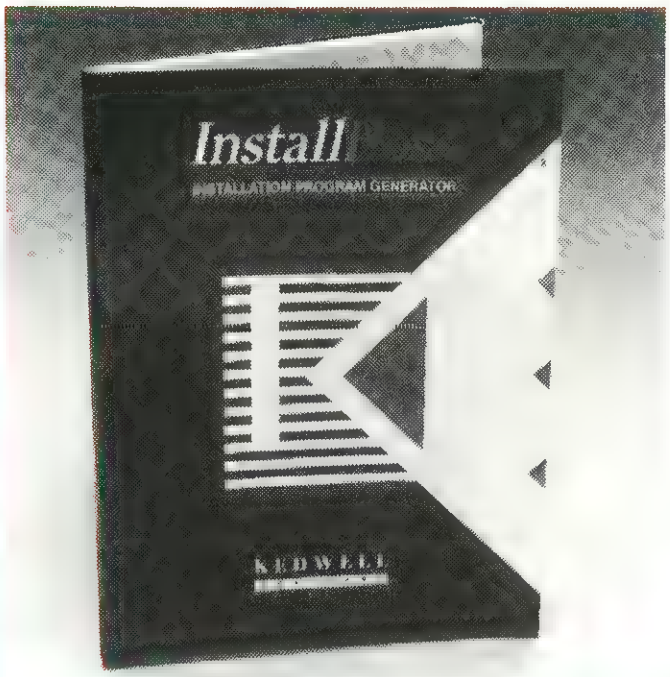
* Prices exclude delivery charges.

CIRCLE NO. 116

```
// connect.cpp - member functions of Packet,
//               Protocol, Connection.
//  - Laine Stump, July 1991, No Rights
Reserved -

#include  "connect.h"

//////////////////////////////////////////
// Members of Packet
int Packet::CkSum()
  { // generate a checksum based on contents
  int ck = ((int) Type) + Seq + Valid;
  for (int ct = 0; ct < Valid; ct++)
    ck += Data[ct];
  return ck;
  } // Packet::CkSum()


//////////////////////////////////////
// Members of Protocol
void Protocol::SendCharCAN(char c)
  { // send a char of packet, check for CAN
  SendChar(c);
  if (c == CAN) SendChar(CAN+0x40);
  } // Protocol::SendCharCAN()

int Protocol::RecvCharCAN()
  { // receive a char, check for CAN
  int tmp = RecvChar() & 0xFF;
  if (tmp != CAN) return tmp;
  // CAN recved, encode following char
  tmp = RecvChar() & 0xFF;
  if (tmp < 0x40) return -tmp; // cmds,
types
  return (tmp-0x40);           // ctrl chars
  } // Protocol::RecvCharCAN()

void Protocol::XmitPacket()
  { // send *Sending out the connection
  SendChar(CAN);
  SendChar((char) Sending->TypeOf());
  SendCharCAN((char) Sending->SeqOf());
  SendCharCAN((char) Sending->SeqOf() >> 8);
  Sending->Again();
  while (!Sending->IsExhausted())
    SendCharCAN(Sending->Get());
  SendChar(CAN); SendChar(END);
  int tmp = Sending->CkSum();
  SendCharCAN((char) tmp);
  SendCharCAN((char) tmp >> 8);
  } // Protocol::XmitPacket()


void Protocol::GrabPacket()
  { // state machine to grab one entire
Packet
  int state = 0;  // idle
  int tmpseq, tmpcksum;
  Recving = new Packet(DATA, PSIZE);
  do {  // get next char
    int x = RecvCharCAN();
    if ((x <= (int) -DATA) && (x >= (int) -
RESET))
      {  // reached beginning of packet
      Recving->SetType((PacketTag) -x);
      Recving->Empty();
      state = 1;
      continue;  // skip switch
      }  // if packet start

    switch (state)
      {
      case 0:  // idle
        break;
      case 1:  // Seq byte 1
        tmpseq = x & 0xFF;
        state = 2;
        break;
      case 2:  // Seq byte 2
        tmpseq += (x << 8);
        state = 3;
        Recving->SetSeq(tmpseq);
        break;
      case 3:  // GetData
        if (x == (int) -END)
          state = 4;
        else if (Recving->IsFull())
          state = -1;  // NAK & restart
        else
          Recving->Put(x);
        break;
      case 4:  // cksum byte 1
        tmpcksum = x & 0xFF;
        state = 5;
        break;
      case 5:  // cksum byte 2
        tmpcksum += (x << 8);
        if (tmpcksum == Recving->CkSum())
          state = 99;  // finished
        else
          state = -1;  // NAK & restart
        break;
      }  // switch (state)


    if (state == -1)
      {  // NAK - bad checksum
      schWaitLibSemaphore();
      Packet *p
        = new Packet(NAK,0,Recving-
>SeqOf());
      ToSend.insert(p);
      schFreeLibSemaphore();
      state = 0;  // restart
      }
    } while ((state != 99) && (!Done));
  }  // Protocol::GrabPacket()

void Protocol::PacketSender()
  { // packet sending process
  tTimerHandle time = schGetTimer();
  while(!Done)
    {
    schWaitLibSemaphore();
    if( !(Sending = ToSend.get()) )
      {  // no packet to send
      schFreeLibSemaphore();
      if (!WaitingACK)
        schSuspend(SendPID);  // sleep
      }
    else if (WaitingACK
        && (Sending->TypeOf() == DATA))
      {  // waiting for ACK, can't send
DATA now
      ToSend.insert(Sending);  // put back
      Sending = 0;
      schFreeLibSemaphore();
      schScheduleNext();     // next process
      }
    else
      {  // either not WaitingACK or !Data
      schFreeLibSemaphore();
      XmitPacket();
      switch (Sending->TypeOf())
        {
        case DATA:
          WaitingACK = Sending;
          schSetTimer(time, TIMEOUT);
          break;
        case RESET:
          SendSeq = Sending->SeqOf();
        default:
          delete Sending;
          break;
        }  // switch
      Sending = 0;
      }  // if a packet was sent
    // check for timeout
    schWaitLibSemaphore();
    if (WaitingACK && schTimeUp(time))
      {  // timed out waiting for ACK, re-
send
      ToSend.insert(WaitingACK);
      WaitingACK = 0;  // ok to send DATA
      }
    schFreeLibSemaphore();
    }  // while(.Done)
  schReleaseTimer(time);
  Done--;
  } // Protocol::PacketSender()


void Protocol::PacketRecver()
  { // packet sending process
  GrabPacket();
  while (!Done)
    {
    schWaitLibSemaphore();
    switch (Recving->TypeOf())
      {
      case DATA:
        if (Recving->SeqOf() == RecvSeq)
          {  // next packet in sequence
          Recvd.append(Recving);
          RecvSeq++;
          }
        Packet *p  // ACKnowledge receipt
          = new Packet(ACK,0,Recving-
>SeqOf());
        ToSend.insert(p);
        schFreeLibSemaphore();
        schResume(SendPID);
        break;
      case ACK:
        if (WaitingACK && (Recving->SeqOf()
              == WaitingACK->SeqOf()) )
          {  // recved ACK, discard packet
          delete WaitingACK;
          WaitingACK = 0;
          }
        delete Recving;
        schFreeLibSemaphore();
        break;
      case NAK:
        if (WaitingACK && (Recving->SeqOf()
                == WaitingACK-
>SeqOf()) )
          {  // recved NACK, resend packet
          ToSend.insert(WaitingACK);
          WaitingACK = 0;
          }
        delete Recving;
        schFreeLibSemaphore();
        schResume(SendPID);
        break;
      case RESET:
        RecvSeq = Recving->SeqOf();
        delete Recving;
        schFreeLibSemaphore();
      }  // switch (Recving->TypeOf())
    Recving = 0;
    GrabPacket();
    }  // while (!Done)
  Done--;
  } // Protocol::PacketRecver()


Protocol::Protocol()
  { // NOTE: Protocol won't work until a
derived
    // class has started up processes which
call
    // PacketSender() and PacketRecver()
  SendSeq = RecvSeq = Done = 0;
  Sending = WaitingACK = Recving = 0;
  Packet *p = new Packet(RESET, 0, 0);
  ToSend.insert(p);  // reset other end
  } // Protocol::Protocol()

Protocol::~Protocol()
  { // kill processes and release memory
  Done = 2;    // indicate done to
Sender/Recver
  schResume(SendPID); schResume(RecvPID);
  tTimerHandle time = schGetTimer();
  schSetTimer(time, 10);
  while (Done && !schTimeUp(time))
    ; // Sender/Recver each do Done-- at end
  schReleaseTimer(time);
  delete Sending; delete WaitingACK;
  delete Recving;
  while (Sending = ToSend.get())
    delete Sending;
  while (Sending = Recvd.get())
    delete Sending;
  } // Protocol::~Protocol()

void Protocol::SendPacket(Packet *ToGo)
  { // dispatch a data packet to PacketSen-
der
  ToGo->SetSeq(SendSeq++);
  schWaitLibSemaphore();
  ToSend.append(ToGo);
  schFreeLibSemaphore();
  schResume(SendPID);  // wake up sender
  } // Protocol::SendPacket()


Packet *Protocol::RecvPacket()
  { // get the next data Packet received
  Packet *tmp;
  do {
    schWaitLibSemaphore();
    tmp = Recvd.get();
    schFreeLibSemaphore();
    } while (!tmp);
  return tmp;
  } // Protocol::RecvPacket()

Packet *Protocol::TryRecvPacket()
  { // return next Packet if ready, else 0
  Packet *tmp;
  schWaitLibSemaphore();
  tmp = Recvd.get();
  schFreeLibSemaphore();
  return tmp;
  } // Protocol::TryRecvPacket()

//////////////////////////////////////////
// Members of Connection

int Connection::Ready()
  { // see if there is a char ready to get '
  if (Coming->IsExhausted())
    {
    Packet *p = TryRecvPacket();
    if (p)
      {  // only delete old if new available
      delete Coming;
      Coming = p;
      }
    }
  return (Coming->IsExhausted());
  } // Connection::Ready()


char Connection::Get()
  { // get one char from Coming
  if (Coming->IsExhausted())
    {
    delete Coming;
    Coming = RecvPacket();
    }
  return Coming->Get();
  } // Connection::Get()

void Connection::Put(char c)
  { // put one char to Going
  Going->Put(c);
  if (Going->IsFull())
    Flush();
  } // Connection::Put()

void Connection::Flush()
  {
  if (!Going->IsEmpty())
    {
    SendPacket(Going);
    Going = new Packet (DATA, PSIZE);
    }
  } // Connection::Flush()
// end of connect.cpp
```

*Figure 3 - CONNECT.CPP*

# Make the best impression with your Boss!




**InstallBoss** generates installation programs for MS-DOS based computers.

**InstallBoss** generates professional installation programs and procedures for your software application.

**InstallBoss** produces programs that are simple and easy to use.

## InstallBoss makes the best impression!

**InstallBoss** is a menu-driven application that creates .EXE installation programs.

The package can be learned in about five minutes and includes sample files to get the user up and running.

As an end-user you require no technical knowledge. You simple type A:Install, and InstallBoss does the rest.

**InstallBoss** uses self-extracting archive files created with a file compression utility.

The program prepares distribution diskettes for simple, hands-free installation by the end-user.

This means both programmers and end-users save time because less technical support is needed.

**InstallBoss:**
- no programming
- modifies config.sys
- modifies autoexec.bat
- no script files
- no licence fees
- selective installation
- on-screen, easy help

Telephone: 010-353-1-2804839
Fax: 010-353-1-2805082

Kedwell Software Europe, PO Box 50, Dun Laoghaire Co. Dublin.

CIRCLE NO. 117

A Packet in memory and a packet going across the Connection are two completely different animals. This translation is handled by the XmitPacket() and GrabPacket() members of the Protocol class. See the definition of Packet in CONNECT.H (Figure 2) for a graphical description. Basically, all special 'command' bytes are preceded by a CAN character (ASCII 0x18); if CAN is encountered as part of the data stream, it is encoded as CAN, CAN+0x40.

Although I would have liked GrabPacket() and XmitPacket() to be members of Packet rather than Protocol, this was impossible, due to the inability to send the address of a member function as an argument to a function. GrabPacket() must call SendChar() which is a virtual member function of Protocol. Since each Protocol (or Connection) has many Packets, it cannot be derived from Packet. Therefore SendChar() cannot be a member of both Packet and Protocol. The result is a bit of mixed functionality between the two classes. Not elegant, but it works.

## Protocol

The Protocol class handles all the details of scheduling packets to be sent, holding packets after they are received, waiting for a packet's ACK after it has been transmitted, and transmitting the ACKs of received packets. This class is the work-horse of the system and, due mostly to the limitations of C++ in a multi-threaded environment, can be quite confusing.

I think of Protocol in three parts: the sender (PacketSender()), the Receiver (PacketRecver()) and the interface (SendPacket() and RecvPacket()).

Protocol::PacketSender() is not directly called, it is set up as a separate process. It continually monitors the list ToSend, looking for Packets to send across the line. When there is a Packet, it sends it using XmitPacket() (which translates the Packet from in-memory representation to line representation). Then, if the Packet is a DATA type Packet, it is saved in WaitingACK until either an ACK is received (indicated by PacketRecver() putting a 0 in WaitingACK) or a time-out occurs. When time-out occurs, the packet is re-sent.

***You think that using some other method will get rid of the complicated stuff. It doesn't. It just delays it until later***

In order to avoid losing track of which packets haven't been ACKed while waiting for the ACK, PacketSender() will not attempt to send any other DATA packets. ACK and NAK packets can and should be sent, however, to avoid deadlock on the line (both sides could be expecting an ACK from the other side, but not sending anything, including ACKs, because a packet was waiting for ACK). To make sure that ACK and NAK packets are always sent as soon as possible, they are put onto ToSend at the head of the list (with insert()), while DATA packets are put onto the tail of the list (append()). This also helps avoid deadlock.

Protocol::PacketRecver() is also set up as a separate process which is constantly reading characters from the line trying to assemble a packet. GrabPacket() is a state machine which knows how a packet should look on the line. The representation of packets is such that the sequence 'CAN [type]' will never be encountered except at the start of a packet. Because of this, GrabPacket() knows that any time it sees such a sequence, it can reset to the start of the packet. This aids in resynching the connection after a burst of errors, much more efficiently than the usual 'idle the line for x seconds' method used by many protocols. (This method of synching packets was borrowed from Zmodem).

The action taken by PacketRecver() upon receiving a packet depends on the packet's type. If a DATA packet is received, it is put onto the Recved list (after checking for proper sequence), and an ACK is inserted onto ToSend. A received ACK packet causes the packet in WaitingACK to be deleted (as it has already been confirmed as sent), and WaitingACK to be set to 0, indicating another DATA packet may be sent. NAK puts the packet in WaitingACK back onto ToSend (using insert()) for retransmission. RESET is a special packet which allows us to get the two ends of the connection into synch concerning sequence numbers.

Note that packets with bad check-sums are handled directly by GrabPacket(); they are never seen by PacketRecver().

```
// test.cpp - test the SerConnection class
//
#include <bios.h>
#include <stdio.h>
#include "connect.h"

////////////////////////////////////
// a descendant of Connection: SerConnection

class SerConnection : public Connection
  {
public:
  tporttype Port;

  SerConnection(tProcess send, tProcess
recv,
    tPortType port, int baud);
  ~SerConnection();

  virtual void SendChar(char c)
    { mdmWrite((unsigned char) c); }
  virtual char RecvChar()
    { return (char) mdmRead(); }

  }; // class SerConnection

SerConnection
  ::SerConnection(tProcess send, tProcess
recv,
                  tPortType port, int baud)
  : Connection()
  { // set up modem
  Port = port;
  mdmInstall(Port, 2048, 2048);
  mdmSetSerial(baud, 8, mdmNoParity, 1);
  // use CTS/RTS handshaking
  tFlowControl FlowCtl;
  FlowCtl.RecvXON = FlowCtl.SendXON = 0;
  FlowCtl.DSR   0;
  FlowCtl.CtrlC = 0;
  FlowCtl.CTS = 1;
  mdmSetMode(FlowCtl);

  // start sender and receiver processes
  SendPID = schSpawnProcess(send, 4096);
  RecvPID = schSpawnProcess(recv, 4096);
  } // SerialConnection::SerialConnection()

SerConnection::~SerConnection()
  {
  mdmUnInstall(Port);
  } // SerialConnection::SerialConnection()

// a serial connection on Com 1
void  far Com1Send(); void  far Com1Recv();
SerConnection Com1(Com1Send, Com1Recv,
                   mdmCom1, 19200);
void  far Com1Send()
  { // dummy routine to start up PacketSen-
der
  mdmSetPort(Com1.Port);
  Com1.PacketSender();  // go to "real" pro-
cess
  } // Com1Send()

void  far Com1Recv()
  { // dummy routine to start up PacketRec-
ver
  mdmSetPort(Com1.Port);
  Com1.PacketRecver();  // go to "real" pro-
cess
  } // Com1Recv()

int main()
{
while (1)
  { // route keyboard to Com1, Com1 to
screen
  if (bioskey(1))
    {
    int ch = bioskey(0);
    if (ch == 0x1B) return 0;
    if (ch == 0x0D)
      Com1.Flush();
    else
      Com1.Put(ch);
    }
  if (Com1.Ready())
    putchar (Com1.Get());
  }
return 0;
} // main()
 / end of test.cpp
```

*Figure 4 - SerConnection, an example descendant of Connection*

## Interfacing with Protocol

DATA packets are sent to `PacketSender()` and received from `PacketRecver()` via `SendPacket()` and `RecvPacket()`. These two functions merely append a packet on `ToSend`, or get a received packet from `Recved`. This is only used for DATA packets. ACK and NAK packets are sent by calling `ToSend.insert()` directly. Received packets other than DATA are automatically handled and disposed of by `PacketRecver()`.

Although physical communication is performed by some members of `Protocol`, the method of doing so isn't known by `Protocol` itself. All members of `Protocol` use the virtual functions `SendChar()` and `RecvChar()` to send characters across the line. These two functions are redefined for each new descendant class. See the definition of `SerConnection` in Figure 4 for an example.

## Connection

Class `Connection` is the simplest of the classes. It puts a character based wrapper around the packet oriented functions of `Protocol`. `Connection::Put()` collects characters in a packet until either the packet is full, or `Connection::Flush()` is called, when it sends the entire packet off to `Protocol`.

`Connection` could be made even more useful by setting up a timer event to flush the current packet after a certain time. This would allow true character based communications, very similar to the MNP protocol used by many modems.

`Connection`, like `Protocol`, is a virtual class. You should never declare an object of type `Connection`. Derive a class from Connection which defines `SendChar()` and `RecvChar()`, and declare an object of the derived type.

## SerConnection

The program in Figure 4 declares a descendant of `Connection` called `SerConnection`. As well as defining `SendChar()` and `RecvChar()`, it defines a constructor and a destructor.

One important function of the constructor is to start up the `PacketRecver()` and `PacketSender()` processes. These must be started by the final descendant class rather than `Protocol` or `Connection` because their constructors are called before those of `SerConnection`. If the processes were started before `SerConnection`'s constructor was called, the virtual function pointer table for `SerConnection` wouldn't be set up yet, and the calls to `SendChar()` and `RecvChar()` would branch to Never-never Land.

***Implementing Connection as multiple processes has shown some holes in the C++ language***

We've almost got a `Connection` up and running now. We just have to talk about...

## The Ugly Part

I said before that `Protocol` was a bit confusing due to limitations of C++. One ugly part is what we saw above concerning starting up the send and receive processes. In an ideal world, a descendant of `Protocol` would not have to know that `Protocol` works as three processes. But because we had to get the virtual function pointer table set up, we had to defer the process setup until `SerConnection`'s constructor.

There's still another ugly point, also concerned with processes. While the process scheduler wants the address of a function to activate as a process, we can't take the address of a member function of an instance of an object, eg `Com1.PacketSender()`. Instead, we must write a short 'dummy' function which calls the member function; the address of the dummy function is given to the process scheduler.

As we must be able to access the object (in this case `Com1`) from the dummy function, the object must be global in scope. This means that descendants of `Connection` cannot be local in scope (although they could be dynamically allocated and pointed to by a global `Connection*`).

`Com1Send()` and `Com1Recv()` are the dummy functions used to start up `Com1.PacketSender()` and `Com1.PacketRecver()`. Their addresses are sent as argument to the `SerConnection` constructor, so that different instances of `SerConnection` can be set up for different ports.

In the end, the declaration of a new type of `Connection` is much more complicated than I would like. It reminds me of what a friend once said about the merits of various different programming methods: 'You think that using some other method will get rid of the complicated stuff. It doesn't. It just delays it until later'.

## Conclusion

X/Y/Zmodem are great for simple file transfers during dial-up communication, but should be forgotten for any kind of automated system with data flowing in both directions.

`Connection` is a good start at an efficient full duplex protocol. It could use some work in the error recovery area. It could also be made more efficient on overseas (satellite) lines and packet switching networks by using packet windowing techniques. File transfer protocols can easily be built on top of `Connection`, and can be designed assuming error free data transmission.

Implementing `Connection` as multiple processes has shown some holes in the C++ language. I've read some blurbs about Concurrent C++. Maybe I should check it out.

EXE

---

*Laine Stump has spent the last several years in Turkey, trying to keep in touch electronically with his techie friends and family in the US. He can be reached from the Mail room of the PC Tech, Inc 'Citadel' BBS at 0101 612 345 4656 (3-2400 8,N,1) as 'Laine Stump' (what else?) Even his mother had to resort to this method to stay in touch with him.*

*The full source code given in this article is available on disk. Send a blank floppy disk to the Editor, following the instructions given on Page 1, column 1. Mark your envelopes 'COMMS'. Alternatively, modem-owning readers may download the material from the Communications room of the PC Tech, Inc BBS system (details as above) during evening hours (USA), or from the `exe` conference on CIX.*

The Solid Link Communications Library *is available from Solid Software Inc, PO Box 36164, Cincinnati, OH 45236-6164, USA.*

# Five go LAN in Chiswick

*Networking is big business and Microsoft wants a piece of the action. Paul Kemp takes a look at LAN Manager 2.0.*

It is reckoned that by 1993 almost 50 percent of all PCs used in the business sector will be networked. That's about 35 million machines. It would therefore seem a reasonable assertion that the local area network (LAN) has finally come of age. The stakes are high but the potential rewards for vendors are great indeed. Software giant Microsoft is pinning its networking strategy on the LAN Manager product range. This article is a guided tour of the major features of LAN Manager version 2.0, including setup and usage.

## Overview

LAN Manager 2.0 is a local area network operating system which provides standard network facilities, such as disk, printer and comm device sharing, and a sophisticated platform for utilising and developing distributed processing (client-server) applications like SQL Server. Tools for managing the security and administration of the network are provided and an extensive API is available to programmers. LAN Manager servers run the OS/2 operating system and workstations can be DOS, Windows or OS/2 (support for Mac workstations should be available before the end of the year). A minimum of 5 MB RAM is required for OS/2 servers, although considerably more is recommended.

The hardware configuration used for this article was as follows: *Server* - A 33 MHz, 486-based machine running OS/2 1.3, with 16 MB RAM, 1 GB disk space and a 16-bit Western Digital Ethercard Plus network adapter. *Workstations* - a variety of 386-based PCs running DOS 5 with a number of different 8-bit Ethernet network adapter cards.

## History

In 1983 and 1984, Microsoft and IBM were designing DOS version 3.0 to support the enhanced hardware features of the IBM PC/AT - 20 MB hard disk, 1.2 MB floppy and CMOS configuration and clock. At this time it was also decided to build in support for local area networking. In the end, DOS 3.0, released in April 1984, only contained part of the intended networking capabilities. It wasn't until version 3.1 came out in July of that year that there was support for a *redirector* which used Server Message Block (SMB) protocols, specified by IBM, and the NetBIOS to allow remote file access.

IBM and Microsoft then went their separate ways (sounds familiar eh?) and two network products emerged: the *Personal Computer Local Area Network Program* (PC-LAN) from IBM and *Microsoft Networks* (MS-NET). The two products differed in that PC-LAN supported non-dedicated servers (servers that could also act as workstations) and interstation messaging. Neither of these features were supported by the Microsoft product, although the two systems were interoperable.

Now comes the confusing bit. After DOS 3.1, Microsoft began to develop, independently, a multi-tasking, protected-mode operating system for the 80286. As it grew, the project was split into two parts: DOS 4 (no, not the one you're thinking of) was a real-mode multi-tasking DOS which was never released as a retail product. But part of the OS was an upgrade of MS-NET to version 2.0 that made it functionally the same as PC-LAN. Phase II was provisionally designated the DOS 5 project and eventually became OS/2, developed jointly by IBM and Microsoft. Part of Microsoft's involvement in the project was the creation of an OS/2 redirector and it was at this time that LAN Manager was born.

LAN Manager version 1.0 was not a retail Microsoft product, rather it became the basis of a number of OEM network products. 3Com (which had helped Microsoft with design and development) was shipped in 1988 and it formed the core of the company's *3+Open* network. Other OEM products, such as Torus's *Tapestry II*, followed. 1990 saw the release of LAN Manager version 2.0 for OS/2, the first retail Microsoft version. The product took advantage of OS/2 V1.2's HPFS and added support for UPS, fault-tolerance and the concept of a 'domain' to group servers. That year also heralded the announcement of LAN Manager/X, a portable version for UNIX, developed jointly by Hewlett-Packard and Microsoft. The LAN Manager/X interfaces were adopted by X/OPEN as the portable networking interface for UNIX.

## Installation

Ay, there's the rub. You remember Basil Fawlty's classic exposition of techno-frustration in the scene where we witness him, fizzing with rage, threatening Sybil's recalcitrant vehicle with a leafy branch ('Right! I'm going to count to three...'). Well, unluckily, hardware and software incompatibility conspired viciously to delay installation, so that scenes not entirely dissimilar were commonplace for well over a week in the *.EXE* editorial office.

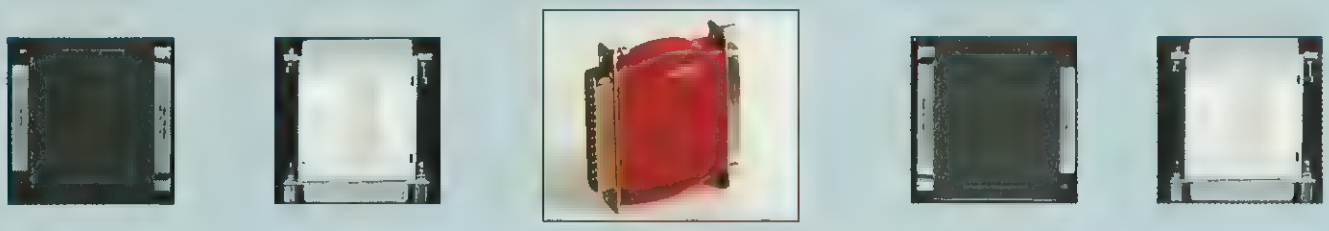
A Novell NE1000 network adapter 'clone' failed to behave itself, so that a host of

| IRQ | XT | AT |
|---|---|---|
| 2 | EGA/VGA | EGA/VGA |
| 3 | COM2 | COM2 |
| 4 | COM1 | COM1 |
| 5 | Hard disk | LPT2 |
| 7 | LPT1 | LPT1 |
| 10 | n/a | unused |
| 11 | n/a | unused |
| 15 | n/a | unused |

*Figure 1 - IRQ Levels of Common Devices*

# SOFTWARE PROTECTION!

## *Can You Spot The Difference?*

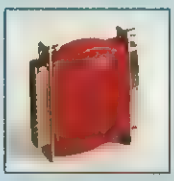


- ☑ High-level security keys
- ☑ Assembler-based, customer specific, encrypted interrogation routines
- ☑ Over 70 languages supported
- ☑ Compatibility - due to 8 years experience & 600,000+ keys sold
- ☑ Reliable on-going support
- ☑ Physically unique keys for each customer
- ☑ MS-DOS, OS/2, UNIX, XENIX, WINDOWS 2 & 3, "MACINTOSH"
- ☑ Transparent operation with most peripherals on market
- ☑ Free language updates
- ☑ Parallel, Serial, Mac ports

## OVER 3,100 CUSTOMERS WORLDWIDE HAVE!

**Electronic Key**

The ideal device for identically produced software packages. Uniquely wired with customer code and a software code. Uses Assembler based program, decryption interface and random values.

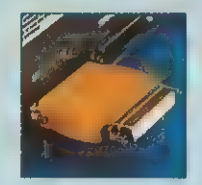


**1 Word Memory Key**

Custom hardware wiring allows the developer total control over information stored in the key. 2 bytes of memory allows several packages to be protected with just the one key.

**31 Word Memory Key**

For multiple software protection schemes. 31 words of 16 bits of non-volatile dynamically programmable memory. Its capacity to store information provides virtually limitless power. Flexible protection scheme can be modified on-site during operation of software package.



**Micro Processor Key**

Provides the ultimate in software security. Not tied to any language or O/S. 8 bit microprocessor powers from RS-232 level. Requires no power supply. For PC terminals, minis, & others using RS 232 C comms. Used on workstations. This key is effectively a computer.

**Macintosh Key**

Extremely powerful & customised protection for the Mac. 31 words available for random storage. Providing unequalled protection the Macintosh Memory Key connects to the SCSI 25 pin port and operates transparently.

## MICROPHAR... The European Standard!

For a no-nonsense informative discussion on how our dongles can better protect your profits, please call us on:
*091-378 9191*

**CLEARSOFT**

CLEARSOFT, Littleburn Ind. Estate,
Langley Moor, Durham DH7 8HG.
Telephone: (091) 378 9191 Fax: (091) 3789393

Please send me full details on your protection systems.

Name: .....

Position: .....

Address: .....

.....

Telephone: .....

Fax: .....

Signature: .....

Clearsoft, Littleburn Ind Est,
Langley Moor, Durham, DH7 8HG
Tel: 091 3789191 Fax: 091 3789393

EXE 9/91

CIRCLE NO. 121

# There Are Now 11 Happy People in Europe. Would You Like to Become the 12th?

Aladdin Knowledge Systems is a world leader in the field of software protection. During the last four years we have appointed HASP distributors in eleven European countries. In each case, success followed rapidly.

Now only the U.K. is missing, and we are looking for an able and ambitious company to fill this gap. If your company has the right blend of marketing talent and technical expertise, please get in touch with us.

NetHASP™, the latest addition to the HASP™ family of software protection systems.


ALADDIN
KNOWLEDGE
SYSTEMS LTD.

15 Beit Oved St., Tel-Aviv, Israel
P.O.Box 11141 Tel-Aviv 61110,
Tel: 972-3-5375795
Fax: 972-3-5375796

CIRCLE NO. 122

unhelpful error messages were generated at logon time (`DOS error 5, SYS0005 The System will not allow access to the file specified`, or `Dos error 53 Network path not found`). These problems disappeared when the card was replaced with a pukka Novell card (and seemed to be OK with other *bona fide* cards such as Western Digital and National Semiconductor). Older versions of 2.0 also appeared to have some problems with the OS/2 1.3 Print Manager. The recent 2.0c LAN Manager upgrade, which adds support for DOS 5 workstations, should dispense with these difficulties. Having said that, one workstation on the network still suffered from corruption of the display when a LAN Manager message was popped up and cleared from the screen. Some network connections to printers were also erratic and occasionally jobs would mysteriously disappear on the way to the printer. My feeling is that the product is a little fussy about the hardware it runs on.

Before plugging in the network card on a server or workstation you may have to configure it so that it doesn't conflict with any other devices on the machine (Figures 1 and 2 show IRQ levels and I/O ports of common devices). It is a little annoying that many 8-bit adapter cards are only configurable in the IRQ range 2-7, which is already crowded (only the 16-bit Western Digital card allowed access to IRQs 10, 11 and 15). Once you've got your hands on the correct hardware and relevant version of the software, installation is actually quite straightforward. On completion of the setup procedure you must first check the PROTOCOL.INI file in the LANMAN directory to make sure that the IRQ and I/O base settings match those of your LAN card (since they are not configurable in the setup) - if they do not, the file can be edited. Servers must then be configured to perform their appropriate role (for example 'primary domain controller' on a network with logon security) as described in chapters four and five of the *Administrator's Guide*. When the network is up and running, workstations can be added quite simply in about half an hour.

## The interface

LAN Manager provides two basic interfaces to its network facilities for administrators and users. The first is the LAN Manager Screen (Figure 3) - a full-screen, text mode, menu-driven interface that pops up when you type `net` at the DOS or OS/2 command line (a functionally extended version is available to administrators by typing `net admin`). LAN Manager commands can also be entered directly at the DOS or OS/2 prompt and have the general syntax: `net` *command parameters*. Remote administration of the network from a workstation is also possible, although the LAN Manager Screen for Administrators is only available from OS/2 workstations.

## Features

Large or diverse networks can be split up into *domains*. A domain is a group of servers and workstations which can be administered as a single logical entity. A user can only logon to a single domain at a time, but can use resources located in other domains if they have the appropriate privileges. A domain running logon security must have a single *primary domain controller* and optionally one or more *backup domain controllers*. The backup servers can validate logons when the primary server is busy or down. They maintain an up-to-date copy of user account information via the server's REPLICATOR service. This service can also be configured to backup any other specified server files. Additionally, OS/2 workstations can be set up as *peer servers*. This allows a PC running the peer service to share disk files and optionally one printer and a communications device (such as a modem).

The *Remoteboot* service supports diskless workstations by allowing a PC to boot OS/2 or DOS from the server's hard disk. In this case, each participating workstation must have a Token-Ring network adapter card with the RPL (Remote Program Load) ROM chip installed. This enables the card to receive startup and configuration software from the server when the workstation starts.

Fault tolerance facilities are built into LAN Manager 2.0 with support for an uninterruptible power supply (UPS) and utilities to enable drive mirroring and duplexing. Other features include interstation messaging; network monitoring through an audit trail; the NETRUN server service for remote program execution; utilities to run programs on a server at a specified time, check storage space and detect errors.

## Security

Access to shared resources is controlled by servers using one of two security modes: *user-level* or *share-level*. Under user-level security, every user of the network must have an account set up on the server. Permissions are then granted for each resource that you plan to share. A user account contains a username and password along with information about how the person can use the server. For instance, you can limit the hours during which a user can access the server's resources and the workstations from which the user can connect to the server. Permissions define who and how users access shared resources. For shared disk resources, permissions can be set at the directory, subdirectory or file level. Permissions can be granted to individual users or groups of users to simplify management.

Other features of user-level security include the following:

- *Logon security*, in which only specified users can logon to the network. Username and password are verified at logon time and a logon script can be created to automatically make network connections to resources and/or run programs when the user logs on.

- *Local security*, available only on 386-based servers using the high-performance file system 386 (HPFS386). Extends user-level security to people working at the server itself. Access to the servers files is strictly governed by permissions that have been set up, regardless of whether LAN Manager is running.

Share-level security uses a single password to limit access to a shared resource, which has only one set of permissions. Any user who can supply the password can use the resource.

## Sharing and using resources

From a PC user's point of view, the most noticeable benefit of being on a network is the ability to access diverse resources which are not directly connected to your computer. These resources can be categorised into three groups:

- Disk resources
- Printers
- Communications devices

The resources themselves must be directly associated with a LAN Manager server (or OS/2 workstation running the peer service). Under LAN Manager, a server's resources are *shared* by any user with an ADMIN privilege level (this can be granted when the user's account is set up). The server can also be set up so that a specified list of resources are shared automatically when the server starts. Resources on a server are shared by assigning a *sharename* which uniquely identifies the resource to users. A user then makes a network connection to the resource by *using* it at the workstation. One can share and use resources in the LAN Manager Screen or at the command line. In a domain with user-level security, each resource is protected by an Access Control List which determines who can use the resource and in what ways.

The following example grants read, write, create and execute permissions to FRED for the C:\UTILS directory on the server named MGR. This directory is shared with the sharename UTILS. FRED then uses this directory as if it were a logical drive on his machine with the drive letter G:

ADMIN (at the server):

```
C:> net access c:\utils /add
FRED:RWCX
C:> net share utils=c:\utils
```

FRED (at his workstation):

```
C:> net use g: \\mgr\utils
C:> g:
G:>
```

Printer and comm resources are shared and used in a similar way to disk resources, except that the devices are shared through logical queues. Many queues (with different priority levels) can be active for a single device and several devices can be accessed via a single queue. Printer queues created in LAN Manager are spooled by the OS/2 Print Manager.

This example shares the printer queue called DOT1 associated with the LPT1: port on the server. FRED then uses the queue as his LPT2: port to print the file TEXT.DAT.

ADMIN:

```
C:> net share dot1=lpt1:
```

FRED:

```
C:> net use lpt2: \\mgr\dot1
C:> copy text.dat lpt2:
```

## Profiles and scripts

Customised workstation configurations can be saved in a *logon profile* on the workstation. These can be loaded from the workstation when required. A special profile called NETLOGON.PRO is reserved for `net use` commands that should be executed at logon time from that workstation. An example of such a file is below:

```
net use h:     \\mgr\data
net use lpt3: \\mgr\laser1
```

In contrast to a logon profile, a *logon script* is stored in a directory on the server and is executed for that user, irrespective of the workstation used. A logon script can be a batch file, an executable file or a profile. A single script file can be associated with many user accounts to provide a more flexible and manageable way to control user access. Below is an example of a batch file logon script:

```
net use s: \\mgr\slib
net use lpt1: \\mgr\dot1
s:\dbase\dbase
```

The common shares set up on a server can be saved to the reserved profile SRVAUTO.PRO and will be executed at startup time.

## DOS 5 considerations

DOS version 5 allows PCs to load memory-resident software, such as LAN Manager, into the *upper memory area* (between 640K and 1024K), freeing the precious resource for use by other programs. This can be a big plus since LAN Manager for an enhanced DOS workstation takes up about 160 KB. Microsoft's DOS 5 comes with upgrades of two LAN Manager workstation files (NETWKSTA.EXE and NETBEUI.DOS) which enable the LAN Manager software to be loaded into free upper memory blocks (UMBs). OEM versions of DOS 5 do not include these replacement files and it is necessary to obtain a LAN Manager 2.0c upgrade to take advantage of this DOS 5 feature.

It is possible to tell DOS 5 to attempt to run LAN Manager entirely in UMBs with minor modifications to the CONFIG.SYS and LANMAN.INI files. More precise control over exactly which components of the network software are loaded into upper memory can be achieved by adding a *loadopts* section to LANMAN.INI. The following example loads the `workstation` service (about 78 KB) into UMBs and all other services into conventional memory:

```
[loadopts]
chknet = low
minses = low
workstation = umb
messenger = low
netpopup = low
```

## The API in brief

The LAN Manager 2.0 API comprises 114 functions callable from C, BASIC, FORTRAN, Pascal and assembler. The Programmer's Toolkit (PTK) for DOS and OS/2 is available separately and provides C header files containing LAN Manager API definitions, constants and structures. So, if you are using a different language, you need to write your own versions of these. Additionally there are about 40 supporting OS/2 API calls (concerned with interprocess communication and printing) that can be used in programs designed to run under that operating system. The naming of functions is 'object-oriented' in the sense that they follow the general form *NetObjectVerb* (hence `NetShareEnum` enumerates the shares on a server).

Functionally, the API can grouped into the following categories: Workstation; Service and Config; Server Admin; Information; Protection; Interprocess Communication (IPC); Printing; and NetBIOS. The LAN Manager Screen itself is a C program written using this API to perform all network management tasks at servers and workstations.

A powerful feature of the API is the ability to perform remote procedure calls. The first argument of many admin functions is a servername. If this argument specifies a valid computername, the function actually executes on the remote server. Provision for writing distributed applications lies in the named pipe and mailslot APIs which allow transparent IPC between programs running on servers and workstations. *Named pipes* are a bidirectional IPC mechanism. They have two sides: The server side creates the named pipe and waits for a client to connect to the other end; the client opens the named pipe as if it were a file using `DosOpen`. Each side obtains a handle and then uses `DosRead` and `DosWrite` to communicate with the other side. By contrast, *Mailslots* are a one-way IPC. The server side of the communication creates a mailslot for receiving messages and the client side can merely write messages to the mailslot. Data is sent and received one message at a time. The server side can read messages from the mailslot, or it can block, waiting for a message to arrive.

## Conclusion

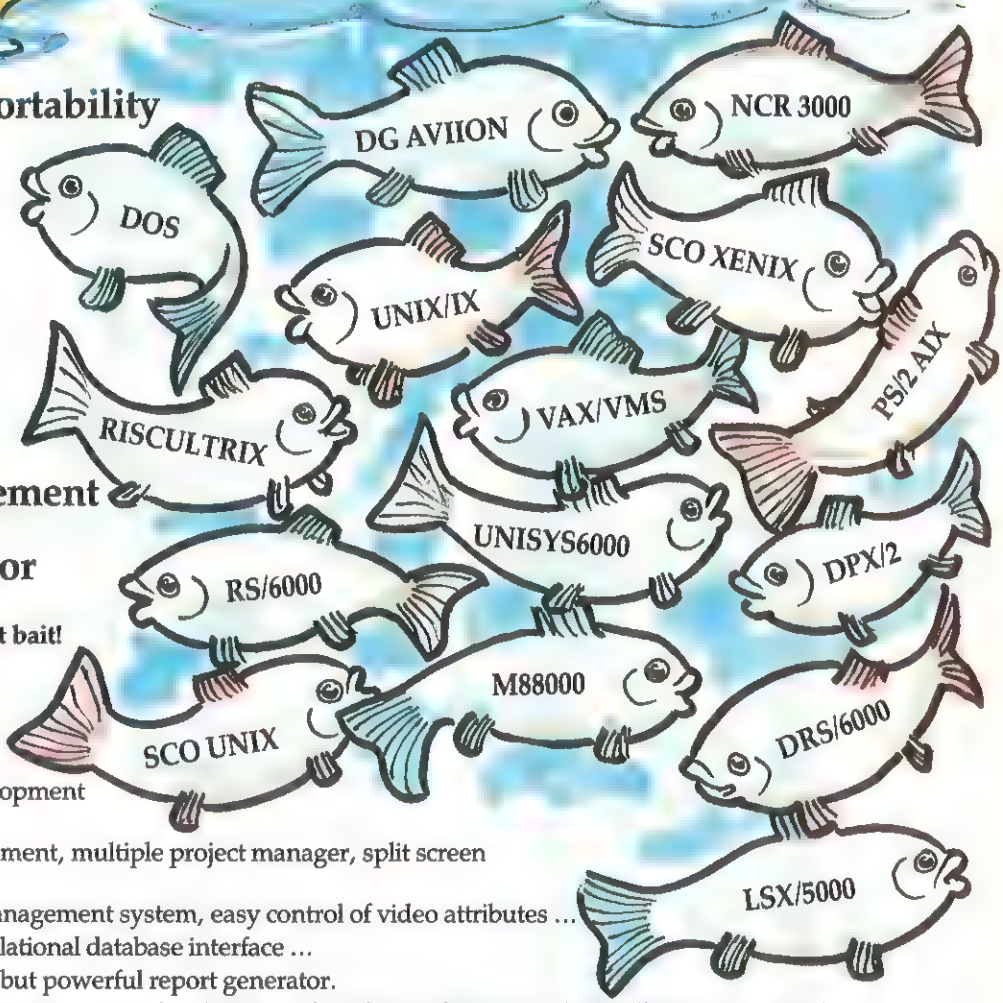
'How did we ever get along without one' I suspect is a phrase commonly uttered in newly networked offices. The fact is that there are very tangible benefits to be realised. Inexpensive PCs can seamlessly ac-

| Possible I/O Base Address | Potential Conflicting Devices |
|---|---|
| 200 | Game Controller (200-20F) |
| 220 | Novell NetWare Key Card |
| 240 | |
| 260 | AT LPT2: (278-27F)<br>PS/2 LPT3: (278-27F) |
| 280 | |
| 2A0 | EGA (alternate) (2B0-2DF) |
| 2C0 | |
| 2E0 | GPIB (adapter 0) (2E1)<br>Data Acquisition (adapter 0) (2E2-2E3)<br>COM2: (2F8-2FF) |
| 300 | Prototype Card (300-31F)<br>3Com EtherLink default (300-31F) |
| 320 | XT Fixed Disk (320-32F) |
| 340 | |
| 360 | AT LPT1: (378-37F)<br>PS/2 LPT2: (378-37B) |
| 380 | SLDC communications (380-38C) |
| 3A0 | BSC communications (primary) (3A0-3A9)<br>Monochrome/parallel printer adapter (3B0-3BF) |
| 3C0 | EGA (primary) (3C0-3CF)<br>CGA (3D0-3DF) |
| 3E0 | Floppy disk controller (3F0-3F7)<br>COM1: (3F8-3FF) |

*Figure 2 - I/O Ports of Common Devices*

# Drop us a line!


- **Complete application Portability**
- **Pop up windows**
- **Pull down menus**
- **4GL RDBM's links**
- **Netware compatible**
- **Codeless screen management**
- **Powerful report generator**




**To catch the right fish, you need the right bait! Ryan McFarland's latest family of RM/.. development tools together with RM's legendary portability means that we can offer you a rather special baitbox;**

**RM/COBOL-85,** high level portable development engine, latest windowing functionality.

**RM/CO,** integrated development environment, multiple project manager, split screen debugging, flexible editing ...

**RM/PANELS,** form design and screen management system, easy control of video attributes ...

**RM/Plus DB** revolutionary transparent relational database interface ...

**RM/Companion,** due for release!! Simple but powerful report generator.

**RM/COBOL-85** for Netware, provides transparent interface between COBOL application and Novell's Btrieve record management system ...

If you are fishing in the Open Systems market and you want to get your customer hooked, then drop us a line at ...

---

**Liant Software Limited**
Ryan McFarland
2 Caxton Street
St. James's Park
London SW1 0QE
Tel: 071 799 2434
Fax: 071 799 2552

Name_____
Company_____
Address_____

Telephone_____ Fax_____

CIRCLE NO. 123

cess (in a controlled manner) shared data, disk space and other resources on more powerful servers, which can offer what is approaching minicomputer performance at a fraction of the cost. Remote drive and printer access, in the setup used for this article, revealed a barely noticeable degradation in performance compared to local connection. The LAN Manager Screen is easy to use but, as a user interface, it lacks the sophistication and functionality of value added products like Tapestry II.

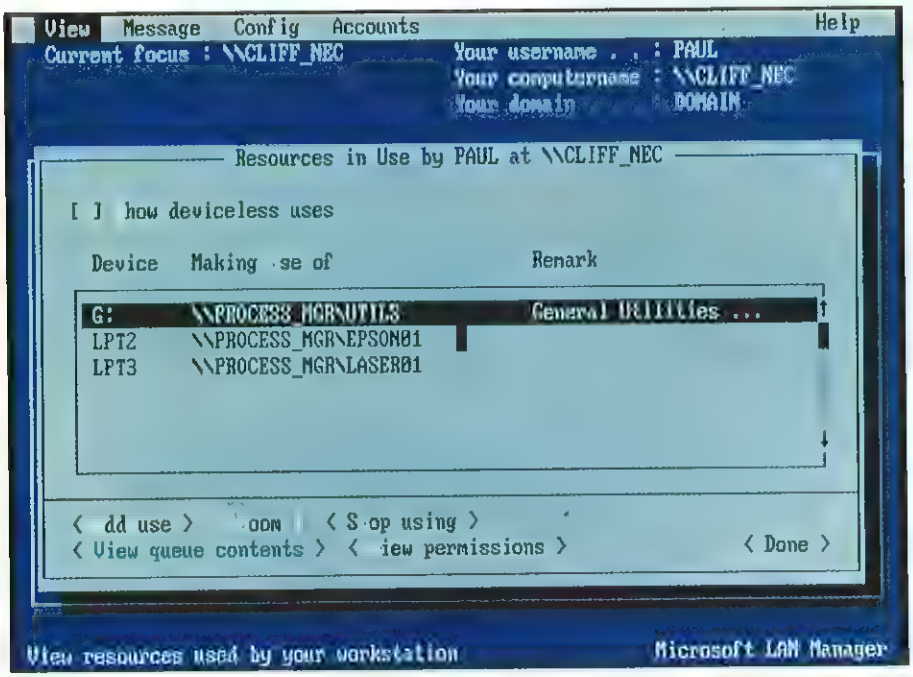



*Figure 3 - The LAN Manager Screen*

LAN Manager is currently being ported to Microsoft's 32-bit Windows NT (New Technology) platform, which it hopes will replace OS/2 in the high-end server market when it is released next year. Although OS/2 support will continue, the company says it will encourage existing users to upgrade to NT. With the huge growth in PC networks predicted to continue, and with the backing of an organisation such as Microsoft, LAN Manager seems destined to be a major player.

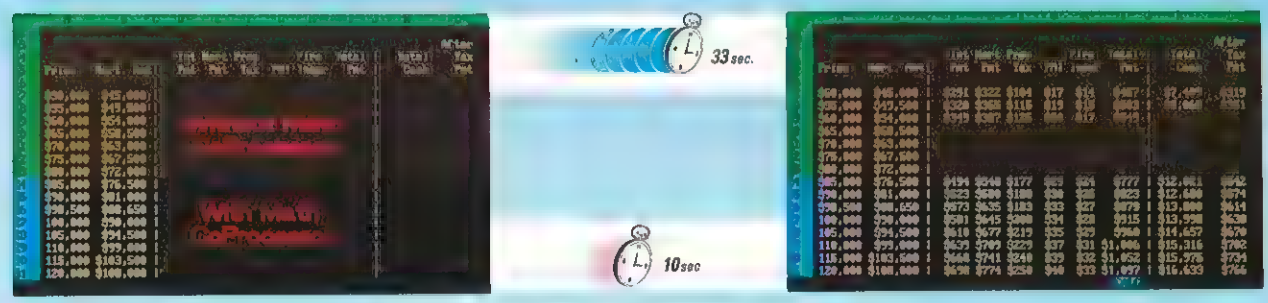
EXE

*LAN Manager 2.0 is priced at £795 for a server pack supporting 1-5 users and bundled with the Sytos Plus file backup manager. Additional 10-user pack: £795. Unlimited-user pack: £4,375 (requires a copy of the basic server installation). The Programmer's Toolkit costs £120.*

*For additional information on LAN Manager/X for UNIX and versions supporting multi-processor platforms (such as Compaq's SystemPro) contact Microsoft on 0734 500741.*

# Lightning Speed for your PC


**WHY** An Intel Math CoProcessor is a chip that plugs easily into an especially designed empty socket you'll find in virtually all personal computers. Intel Math CoProcessors offload the microprocessor in your system to speed through cumbersome numeric calculations. The Intel Math CoProcessor has numeric functions that can complete, in a single instruction, steps that might take the microprocessor hundreds of calculations. The result is that spreadsheets, business graphics, accounting, database, and CAD programs – over 2100 applications in all – run up to five times faster than they would without an Intel Math CoProcessor. Many people think that only scientists, engineers, and programmers need the extra speed Math CoProcessors deliver. This simply isn't true. An Intel Math CoProcessor can be a tremendous help to the business user, too!




*Lotus 1-2 -3 Release 3.1: Benchmarked on an IBM PS/2 Model 55SX 16 MHZ system with Intel387™ SX- 16 Math CoProcessor. Test condition was a Mortgage Payment Schedule.*

IBM PS2 is a registered trademark of International Business Machines

## 300% SPEED IMPROVEMENT FOR 10% OF THE COST OF YOUR PC
## - SAVE TIME - SAVE MONEY - ASK FOR INTEL

# 0256 707386

BYTECH


intel

UPGRADE TECHNOLOGY AT ITS BEST

*A division of Bytech Components Ltd. and a member of Electron House Plc.*

CIRCLE NO. 125

# FOR THE ULTIMATE IN CONNECTIVITY

## SUN PC NFS VER 3.5

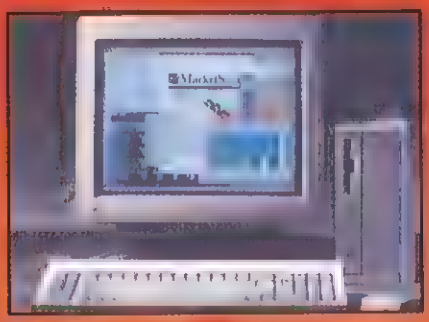
- ★ PCs have direct access to file systems managed by a wide range of operating systems
- ★ 100% MS Windows compatibility
- ★ Up to 14 network devices
- ★ Print server using PC NFS LPD
- ★ VT100 Terminal Emulation

## E92+

TEL: 0634 711700
FAX: 0634 712700

## TCP/IP For DOS

- ★ Full function TCP/IP protocols – TCP, UDP, IP & ARP
- ★ TCP/IP Telnet using H19, VT52 and Ansi X.364
- ★ Standard TCP/IP FTP Program
- ★ Berkeley RSH and ReXec



**ANY OPERATING SYSTEM WITH TCP/IP OR NFS SUPPORT**

Locus Computing Corporation
PC-Interface™ for DOS

- ★ Transparent file & resource sharing
- ★ Easy to configure & use
- ★ Terminal Emulation VT220/VT100
- ★ Remote command execution
- ★ Low memory requirement – 50Kb RAM

All trademarks acknowledged.

Also available:
Complete range of networking cards

HUMMINGBIRD COMMUNICATIONS LTD

## HCL-eXceed family

- ★ Turns PCs into X terminals
- ★ Full X11.4 functionality
- ★ Menu driven configuration with editors & compiler built in
- ★ Auto Font Builder
- ★ Motif, Open Look, UWM, TWM compatibility
- ★ MS-Windows concurrency
- ★ Copy & paste between X and MS-Windows

CIRCLE NO. 126

# A Personal Face for the Mainframe

*Remote access to Mainframe applications and data is a major business requirement. Douglas Burns describes one of the more elegant and reliable ways in which this can be achieved.*

Increasingly, corporate expectations of their computing resources include the ability to share information and applications between different hardware platforms. Although there is no doubt that emerging communications media such as optical fibre and an increase in the availability of client/server applications are the best ways forward, more traditional techniques still provide some scope for the creative programmer. One of the most simple and cost-effective methods of connecting dissimilar machines is to run a Terminal Emulation and File Transfer package, using a low-cost Asynchronous Modem for dial-up access (or possibly a V42-spec modem, if you're lucky!).

This article describes one particular way of developing programs which extend the capabilities of this technique, resulting in elegant and reliable applications. I have limited the scope to one type of communication - IBM PC Compatibles (running MS-DOS) to IBM Mainframes - for two reasons. First, this is probably the most common connection requirement. Second, it would be impossible to cover more than one type of connection adequately.

## IBM communications

IBM supplies a wide range of different hardware platforms. In order to integrate these diverse products into a unified network, IBM has created certain proprietary standards which can be applied to all of them. The one which we will be concerned with in this article is the 3270 Terminal Definition. All IBM systems that allow terminals to be connected to them support this definition, which is used by a family of workstations - the 327x range, and most IBM Mainframe applications support them.

The terminals work as follows. The mainframe sends a screen-full of text, with embedded escape sequences for display formatting. The terminal then assumes control, and has enough local processing capability to allow the operator to move through fields on the screen and insert or delete data. When the operator has finished his entry, and wishes to submit it to the mainframe for processing, he presses one of the 'Attention' keys, which include Enter, Clear, the PFn function keys, and the PAn function keys. Depending on configuration, the terminal will then send either all information on the terminal screen or, more commonly, just the modified fields.

In keeping with this form of terminal I/O, most IBM applications have menu-driven, fill-in-the-blanks user interfaces. One area of potential difficulty is when the operator has just pressed an Attention key, and continues to type. This is precluded by the locking of the Terminal for a short period of time. This state is identified by an X-SYSTEM message in the Operator Information Area, which is described briefly in Figure 2.

PC users who require access to an IBM Mainframe will require some sort of communications package which provides IBM 3270 Terminal Emulation facilities. There are quite a variety of these on the market and your choice will depend on a number of factors, such as the communications options available, the ability to install shared communications on a LAN, whether or not it should be a TSR and so on. Personally, I have used three of these packages, all of which are remarkably similar in capabilities and style. The first two - IBM PC 3270 Emulation Program and Novell's Netware 3270 LAN Workstation - are commonly used in a LAN environment to provide 3270 Terminal access to a number of users. The third, which will be the subject of this article, is IBM's PC/Host File Transfer and Terminal Emulation program (FTTERM). This TSR program allows you to flick screens between a live 3270 session established through an Asynchronous Modem, the parent process (usually COMMAND.COM), and the FTTERM menu. It supports a fair range of communications options, including the normal word format options, XON/XOFF and CTS/RTS flow control, and allows you to employ shared communications on a LAN, if you use a Network Asynchronous Communications Server (NACS) package.

## Making life easier for users

Let's be honest; that's what it's all about. I could write a book, never mind an article, on the subject of improvements which could be made to a PC/Mainframe connection, but will limit myself to a handful :-

- Facilities for unattended operation.

For reasons of lower telephone call charges, increased mainframe response speed or simply to allow maximum usage of a PC during working hours, it may be desirable to have the PC dial-up the Mainframe, Logon and perform certain tasks without the presence of a human operator being required.

- An improved user interface.

The 327x range of terminals are able to provide a full-screen, 'fill-in-the-blanks' type of user interface which, by terminal standards, is a fairly pleasant one to work with. Unfortunately it still falls far short of the expectations of the modern user. It is shoddy to have a user select the 'Transfer File' option from your pull-down menu, only to be confronted by a screen full of Hayes AT commands. There is a great deal of information displayed during a dial-up session which is extraneous to Joe User's requirements. As if all that isn't enough, some mainframe programmers' understanding of user interface design, even allowing for the restrictions imposed on them, are archaic. As everyone knows, a PC has much better screen-handling capabilities than a text-based terminal, so why not let the PC provide the user interface? Your program could take whatever the user inputs and transfer it into a mainframe terminal session which is running in the background. This allows you to provide a Windows or CUA interface to your existing mainframe applications for the benefit of all your PC users, without any expensive mainframe redevelopment.

- A reduction in on-line time.

When users need to fill in a data form on-screen, they will typically take their time, moving backwards and forwards through the entry fields, correcting any errors which they have made as they go. This ties up a dial-in line, causing a waste of communications resources (particularly the user's telephone line - BT makes enough money as it is). A useful facility would be to allow users to enter their data off-line, so that they can work at their own speed. The PC program could then validate the data, dial-up the mainframe, and fill in the mainframe form using the user-supplied data. A mainframe session would only last as long as is absolutely necessary.

- Automatic detection and correction of session errors.

As I mentioned earlier, the 3270 Terminal Definition has a field-based display. Should the user try to type anything into an area of the screen which is not considered a data entry field, the terminal session (and hence the keyboard) will be locked, and the user will have to explicitly release it using the RESET key. It would be far better if the cursor was moved to the nearest valid data entry field, without locking the session.

- Simplified File Transfer procedures.

Most File Transfer facilities which you will want to offer to your users will transfer the

| *Name* | *Action* | *Function Number (func)* |
|---|---|---|
| **CONNECT PRESENTATION SPACE** | Connects a Presentation Space to the API mechanism, making the PS available for use by API functions. Applies all parameters set using the SET SESSION PARAMETERS function (9). | 1 |
| **DISCONNECT PRESENTATION SPACE** | Disconnects Presentation Space, allowing an alternative Presentation Space to be connected. | 2 |
| **SEND KEY** | Sends a string of keystrokes, up to 255 characters in length, to the Host-Connected Presentation Space. | 3 |
| **WAIT** | If an XCLOCK or XSYSTEM state exists on the Host-Connected Presentation Space, this function will wait up to one minute until it clears. If it does not clear, the function returns an error code. This allows you to produce applications which wait until the mainframe is ready before proceeding, without having to worry about timing variations. | 4 |
| **SEARCH PRESENTATION SPACE** | Searches the Host-Connected Presentation Space for the specified string. In most versions of EHLLAPI, you may specify the start position for the search, although FTHLLAPI does not allow this (it always searches the whole screen). | 6 |
| **SET SESSION PARAMETERS** | Series of variable options (for example QUIET, NOQUIET, CONPHYS, CONLOG), which are specified as a comma-delimited list in a string passed to the function. These parameters will become effective the next time that CONNECT PRESENTATION SPACE function is called. | 9 |
| **RESERVE** | Lock the keyboard. | 11 |
| **RELEASE** | Unlock the keyboard. | 12 |
| **COPY OIA** | Copy the Operator Information Area to a pre-defined string, so that you can check for system errors, etc. | 13 |
| **PAUSE** | Generate a real-time pause measured in half seconds. | 18 |
| **RESET SYSTEM** | Reset the HLLAPI Interface to a pre-defined, known condition. Generates a specific error if the HLLAPI interface does not exist, so you can use this function to verify that the loading procedure has been successful. | 21 |
| **RAISE SESSION DTR** | Raises the DTR line on the modem, allowing FTTERM to establish a connection. | 81 |
| **DROP SESSION DTR** | Drops the DTR line on the modem. This will kill any existing session. | 82 |
| **UNLOAD FTTERM AND FTHLLAPI** | Guess what this does! This has always been successful in my experience, provided that the two components were the last thing loaded. | 84 |
| **SEND FILE** | Uses the IND$FILE-compatible SEND.COM program supplied with the package. Be careful with this, and the next function, as the transfer file location is drive specific! You must supply a drive identifier when calling these functions. | 90 |
| **RECEIVE FILE** | Uses IND$FILE-compatible RECEIVE.COM program. | 91 |

*Figure 1 - Basic FTHLLAPI Functions*

UNIPALM TRAINING SERVICES

## TCP/IP

Now an accepted standard, TCP/IP is the language of UNIX, the standard operating system now supplied by all major computer vendors. Our lab-style course gives you plenty of hands-on experience, using products such as FTP's PC/TCP, building up a fast, clear understanding of TCP/IP.

## NFS

Another open systems connectivity standard, now used and accepted industry-wide. We can teach you, using products such as Sun Microsystems' PC-NFS, to understand the concepts and use and administer NFS systems effectively.

## X WINDOW SYSTEM OVERVIEW

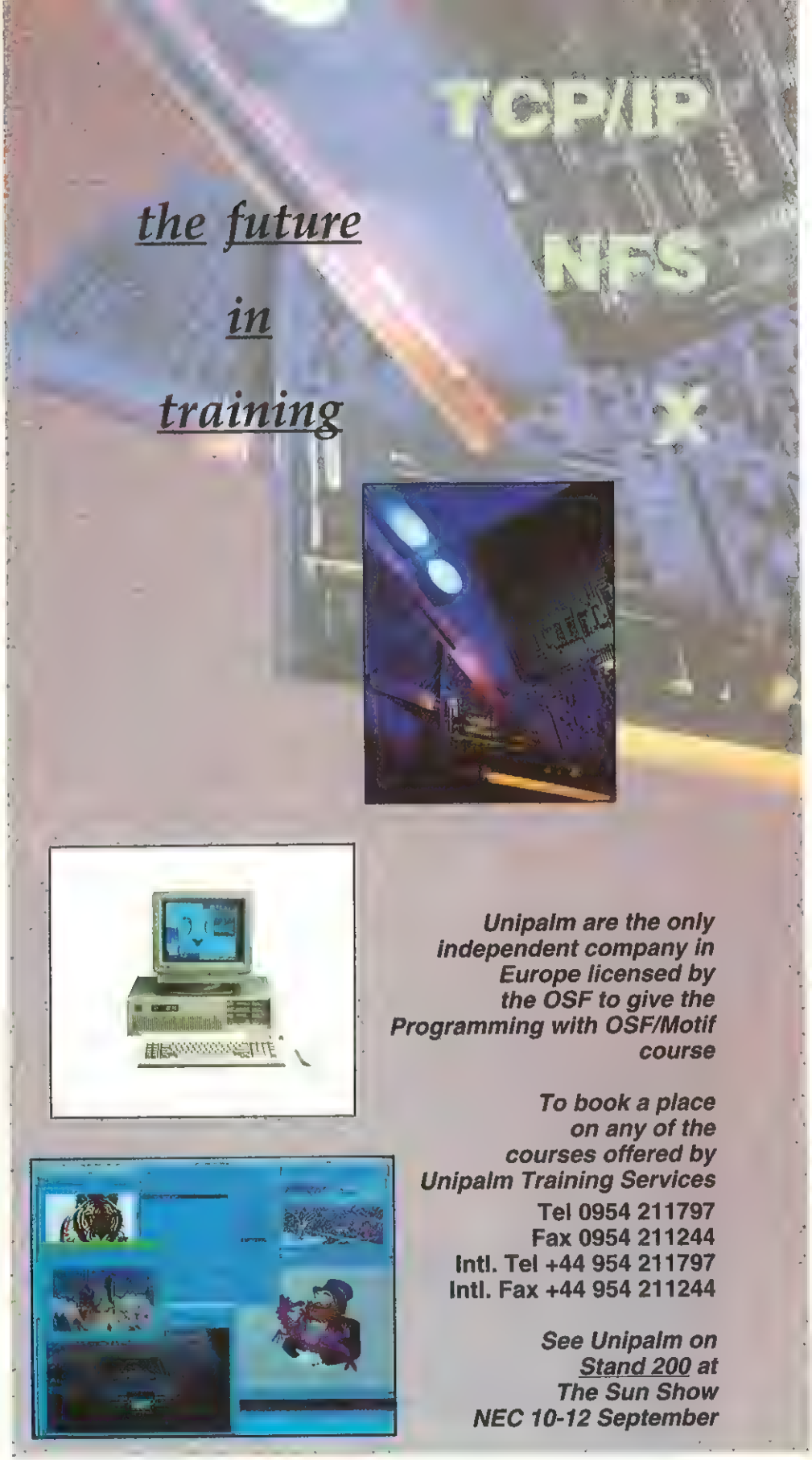
All major software vendors are thinking X now - it's playing a major part in the development of distributed applications. We used our many years of X development, support and marketing experience to develop an X Overview which gives a quick, easy understanding of the principles of X, so you can exploit its power too.

## X ON PCS

To provide cost-effective additional X machines, many people use PC-based X servers to convert existing PCs to X terminals. Our PC-based X training includes plenty of hands-on experience – as user and administrator – using products such as PC-Xview from Spectragraphics, and IIM's Windows 3-compatible X server, X11/AT.

## OSF/MOTIF AND OPEN LOOK

For people developing X applications, or simply wanting an in-depth understanding of the programming principles behind X, we are the only company in Europe licensed to offer the week-long OSF/Motif programming course. We can also help Sun Open Look X users to get started.




*the future*

*in*

*training*







***Unipalm are the only independent company in Europe licensed by the OSF to give the Programming with OSF/Motif course***

***To book a place on any of the courses offered by Unipalm Training Services***

**Tel 0954 211797**
**Fax 0954 211244**
**Intl. Tel +44 954 211797**
**Intl. Fax +44 954 211244**

***See Unipalm on Stand 200 at The Sun Show NEC 10-12 September***

Unipalm Limited 145-147 St Neots Road Hardwick Cambridge CB3 7QJ England
Tel 0954 211797 Fax 0954 211244 Intl Tel +44 954 211797 Intl Fax +44 954 211244

Unipalm

same file. The user must generally select a File Transfer option, then supply the source and destination filenames and, should the transfer fail, the error reporting is not particularly geared to a user's understanding. All of this could be hidden from the user, so that they simply have to select a 'Transfer End of Month Accounts Data' option, for example.

There are two methods which can be used to implement the above improvements. The first is to use a 'Software Robot' utility, such as Automator, mi or Via. The software robot (we will use Automator for our example) is run and sets about trapping all the critical interrupts such as the timer click, key-press and Video BIOS interrupts, then Terminates and Stays Resident. You then run a program which you have written using Automator Command Language, which loads the Terminal Emulator, and sends artificial keystrokes to flick to the terminal screen. Your program sends a keyboard sequence, using Automator, to initialise the modem. If you have written the program well, it will search for an OK response in a window which you have defined to Automator. This process continues, as you instruct Automator to send keys, search for on-screen messages or perhaps pause for a fixed interval at some stages.

Automator is a tremendous all-rounder, allowing you to program automatic control of almost any application, but it does have flaws. First, it traps an enormous amount of interrupts, cannot be unloaded cleanly and, while its behaviour is remarkably good, in my experience it is not perfect. Second, it obviously can not be expected to modify the internal behaviour of another application. If a procedure is ugly or long-winded when performed by a user, it will be the same with Automator (except that Automator is a faster typist!). Automator will only allow you to do what a skilled user could do.

## The HLLAPI solution

The second method is to use an API specifically designed to operate in partnership with your chosen Terminal Emulation software. In order to facilitate the development of terminal emulation enhancements, IBM supplies an API - the Personal Communications/3270 Entry-Level High Level Language Application Program Interface (try saying that at 1.30 on a Saturday night/Sunday morning!), otherwise known as EHLLAPI, or simply HLLAPI ('hi-lappy'). This API allows the developer to call many of the terminal emulator's internal functions using a basic, consistent calling interface. It is available in a number of versions, designed to work with different IBM packages, and you will also find that most other suppliers of 3270 emulation products provide HLLAPI support. For example, in this article I used FTHLLAPI, which is supplied with the aforementioned FTTERM. All versions are based on the same basic calling procedure and available functions, which are described in the EHLLAPI Programming Reference, and any variations from this standard should be documented in the terminal emulator's documentation.

The API consists of two components - a TSR program which accesses the relevant object code in the emulation software, and a linkable object module which allows your program to call the TSR. IBM refer to this object code file as a Language Interface Module (LIM), and supply LIMs for C, FORTRAN, Pascal, COBOL, BASIC and Assembler. There are also full instructions for writing your own customised LIM. The particular LIM used in this article is for the Small Memory Model of Microsoft C V5.1.

All HLLAPI function calls are made using the same mechanism, which is to call the function contained in the LIM for your chosen language compiler, eg

```
HLLC(func, data, len, rc);
```

where `func` is a two-byte integer which gives the API function number which you wish to call, `data` is a string which is used as a general-purpose data-passing variable, `len` is a two-byte integer containing the length of the `data`, and `rc`, another two-byte integer, returns the result of the function call (but may also be used to pass a terminal screen position to the API). In general, the `data`, `len` and `rc` parameters are passed by reference, and the `func` parameter by value.

The FTTERM version, FTHLLAPI, differs from the other HLLAPI versions in a number of respects. First, it only allows you to have one live session on the host computer to which you are connected, where other HLLAPI versions allow multiple, switchable sessions. Second, as FTTERM is designed to be used as a TSR, there is an API function to remove both FTTERM and FTHLLAPI from memory. Third, two extra functions are included to raise and drop the DTR line on the modem, which FTTERM uses as a modem status flag. Finally, the API treats the complete Host Presentation Space as one long string, with no field attributes. The combination of these differences makes FTHLLAPI an easier subject to address in a short article.

## Putting it all together

A simple FTHLLAPI Application is listed in Figure 3. All the program does is load the Terminal Emulation software and API; initialise the API to a usable state, dial-up a remote mainframe and log on; transfer the file TEST.DAT to the mainframe; log off; then close everything down gracefully. I've left out all error-checking to keep it short and sweet.

The action of the program is reasonably self-evident. The first line loads both the emulation software and the API and redirects Standard Output to a temporary file, to maintain a tidy screen display. The function `API_RESET` ensures that the emulation software and API are loaded and initialised to a known state (the function will return an error if both components are not loaded). `API_RAISE_DTR` instructs FTTERM to set the DTR line on the modem High, allowing FTTERM to connect to the mainframe.

The next line calls the 'connect presentation space' `API_CONNECT` function. This is an extremely important FTHLLAPI function, as 50% of the other FTHLLAPI functions require this call to have been made first. An explanation of the concept behind

| | |
|---|---|
| **LIM** | (short for Language Interface Module) is the linkable Object Module which is supplied with the API, one version being supplied for each compatible language. This module handles the different language calling conventions. If a LIM is not supplied for your particular compiler, instructions are given to write your own. |
| **OIA** | (short for Operator Information Area) is the status line displayed at the bottom of the screen. Can be accessed using the HLLAPI function COPY OIA (13). |
| **Presentation Space** | is the term used by IBM to describe a virtual terminal screen. EHLLAPI supports eight presentation spaces, of which one can be connected for use by the API functions, using the CONNECT PRESENTATION SPACE function (1). This Presentation space is then referred to as the Host-Connected Presentation Space. Some HLLAPI functions operate in the Host-Connected Presentation Space by default. |
| **PSID** | (short for Presentation Space ID) is a single character, from A to H, which identifies the Presentation Space in which a function is to operate. |

*Figure 2 - Glossary*

# System Science

• Software • Specialists • Software • Specialist • Software • Specialists • Specialist • Software • Specialists •

## Unix

**SCO** Unix Op. Sys **SCO** Unix Dev Sys.
**SCO** Xenix 386 **SCO** Xenix 286
**SCO** Open Destop **SCO** ODT Dev Sys
**SCO** TCP/IP **SCO** NFS
**Interactive** Unix Op Sys
**Interactive** Architect Series.
**Informix** 4GL & SQL **RM** Cobol-85
**LPI** Compilers **C Scape** Screen Lib
... many more tools and applications

## Mathematics

Derive - The Mathematical Assistant
Mathematica 386/387 (ver 2.)
Mathematica MAC/ENH
Mathematica for Workstations
MathCad for Windows (new) GrafTool
What's Best Lindo
Tex Chiwriter Sci WP

## Windows Development

(see Microsft and Borland)

| | |
|---|---|
| CASE:W Corp. (with FREE SDK) | £995.00 |
| Smalltalk/V Windows | £295.00 |
| Drovers Toolkit Windows | £175.00 |
| Multiscope Windows Debugger | £295.00 |
| WindowsMaker Prof | £795.00 |
| Zinc Interface Lib | £175.00 |
| C++ Views | £225.00 |
| Win++ by Blaise | £150.00 |

## Pascal

(see Borland)

| | |
|---|---|
| TopSpeed Pascal SE | £115.00 |
| Object Professional | £110.00 |
| Btree Filer MU | £120.00 |
| Asynch Professional | £89.00 |
| Blaise Power Tools Plus | £89.00 |
| Blaise Turbo Asynch Plus | £115.00 |

| | |
|---|---|
| PowerBasic | £89.00 |
| many Basic libraries | |
| PC Logo | £60.00 |
| LMI UR/Forth | £295.00 |
| Smalltalk V/DOS | £69.00 |
| muLISP 90 | £225.00 |
| RM Cobol-85 | £call. |

## Microsoft

| | |
|---|---|
| Microsoft C 6.0 | £225.00 |
| MS Fortran 5.1 (new ver) | £220.00 |
| MS Windows SDK | £295.00 |
| MS Win SDK & C6.0 | £425.00 |
| MS Basic Compiler 7.1 | £225.00 |
| MS Cobol 4.0 (DOS & OS/2) | £399.00 |
| MS Macro Assembler 6.0 | £75.00 |
| MS Visual Basic | £99.00 |
| MS Quick C 2.5 | £60.00 |
| MS Quick C & Assembler | £95.00 |
| MS Quick Basic | £60.00 |
| MS Quick Pascal | £60.00 |
| MS Source Profiler (new) | £49.00 |

***Authorised Languages Dealer***

## Borland

| | |
|---|---|
| Borland C++ (Dos & Windows) | £229.00 |
| Turbo Pascal Windows | £115.00 |
| Turbo C++ | £59.00 |
| Turbo Pascal 6 | £75.00 |
| Turbo Pascal 6 Prof | £150.00 |
| Turbo Debugger and Tools | £99.00 |
| Paradox Engine | £275.00 |
| Paradox 3.5 | £445.00 |

***Authorised Languages Dealer***

## Editors

| | |
|---|---|
| Professional Editor (SPE) | £195.00 |
| Brief 3.1 (with Mouse) | £195.00 |
| Kedit (Xedit for PC) | £110.00 |
| Epsilon | £175.00 |

## Configuration

| | |
|---|---|
| PVCS for DOS | £345.00 |
| PVCS 5 User | £1295.00 |
| PVCS Config Builder (PolyMake) | £129.00 |
| MKS RCS 5.1 (with MKS Make) | £155.00 |

## Misc

| | |
|---|---|
| .RTPatch for DOS, Win & OS/2 | £245.00 |
| .RTLink Plus | £295.00 |
| Plink 86 Plus | £295.00 |
| MKS Toolkit (Unix Tools) | £155.00 |
| Personal Rexx | £110.00 |
| Ghost (software testing) | £145.00 |
| Flowcharting III | £150.00 |
| C Programmers Toolbox | £155.00 |

... and many more tools and utilities

## C++ Compilers

| | |
|---|---|
| Zortech C++ DOS & Windows | £225.00 |
| Zortech C++ Dev Ed (incl OS/2) | £360.00 |
| Zortech C++ Sci & Eng | £540.00 |
| Zortech C++ Video (6xVHS) | £275.00 |
| JPI TopSpeed C++ SE | £115.00 |
| JPI TopSpeed C++ PE | £175.00 |
| JPI TopSpeed C++ Pro Dev. | £229.00 |

## C Compilers

see Microsoft

| | |
|---|---|
| Watcom C 8.0 | £245.00 |
| Watcom C386 8.0 PE | £775.00 |
| Aztec C 86 (with ROM support) | £265.00 |
| JPI TopSpeed C PE | £175.00 |

## C Datafile

| | |
|---|---|
| CodeBase 4.2 | £175.00 |
| CodeBase ++ | £175.00 |
| Ctree Plus | £345.00 |
| Btrieve for DOS | £345.00 |

## C Communications

| | |
|---|---|
| Essential Comms | £195.00 |
| Greenleaf CommLib | £195.00 |
| C Asynch Manager (Blaise) | £115.00 |

## C Screens

| | |
|---|---|
| CScape with Look & Feel | £365.00 |
| Vermont Views | £325.00 |
| Panel Plus II | £275.00 |
| Vitamin C 4.0 | £245.00 |

## C Graphics

| | |
|---|---|
| Essential Graphics | £245.00 |
| PCX Programmers Toolkit | £165.00 |
| Halo | £195.00 |
| Ingraf | £195.00 |

... many more libraries

| | |
|---|---|
| Pharlap 386 DOS Ext SDK | £325.00 |
| Pharlap 286 DOS Ext SDK (new) | £325.00 |

## Fortran Compilers

| | |
|---|---|
| Lahey F77L | £375.00 |
| Lahey F77L-EM/32 with OS386 | £845.00 |
| Salford FTN 77-386 | £765.00 |
| Salford FTN77-486 | £875.00 |
| RM Fortran with Forte | £475.00 |

... many Fortran Libraries

- **Prices are exclusive of VAT.**
- **Prices include delivery to GB.**
- **Prices are subject to change.**
- **VISA, Access and Mastercard welcome with telephone orders.**

**3-5 Cynthia St**
**London N1-9JF**
**Fax: 071 837 6411**

# 071 833 1022

• Software • Specialists • Software • Specialist • Software • Specialists •

# Asynchronous Point-to-Point Communications or Telephone-polling Multiple Computers

## XOREN COMMS SOFTWARE– Software with ten years' pedigree

Thoughout the 1980s, Xoren Computing pioneered the development of inexpensive yet reliable communications software. Today, the XOREN IPL-11 range is renowned for enabling automated file transfer between any quantity and combination of the major minis, micros and PCs in configurations which will become more and more commonplace through the 1990s.

### Leading with Automated Connectivity

Although many other communications packages can asynchronously link PCs together, very few besides IPL-11 will orchestrate *fully automated connectivity* between multiple PCs and hosts.

Automated operation with IPL-11 gives users the freedom to perform unattended data communications 24 hours a day. Computers with modems operating up to 9600 baud can be scheduled to dial up remote computers over ordinary telephone lines day or night to send and receive files. Any IPL-11 version can link and transfer files to any other IPL-11 version, regardless of the computers' operating system. Once the files have arrived safely, telephone lines are disengaged.

### Leading with Customer Support

Key to the success of the IPL-11 range is an emphasis by Xoren on customer support.

Because communications is notorious for being a difficult application to implement, Xoren offers installation and support services to ensure the software can operate according to the *user's* requirements.

This principle has made IPL-11 communications software the obvious choice for UK and European companies. Because IPL-11 is a British product, developed and supported in the UK, Xoren is strategically placed to support customers when they need support most - during regular business hours.

Call or write to Xoren today - the most practical solution to automated data communications has always been right on your doorstep.

### XOREN IPL-11 Developments over a Decade

**1980:** Xoren launches IPL-11. Features include peer-to-peer file transfer under operator control, CRC error-checking on individual packets and queuing of file transfer commands.
*Original versions introduced for RSX-11M and VMS for DEC PDP-11 and VAX*

**1981:** Remote activation facility incorporated allowing file transfers when remote computers are unmanned.
*RSTS/E, RT-11 and RSX-11M PLUS versions released for DEC PDP-11.*

**1982:** Terminal emulation facility introduced enabling the use of a terminal on a local computer as a terminal on a remote computer thereby allowing control of file transfer sessions from a single terminal.
*TSX PLUS version released for DEC PDP-11.*

**1983:** Option to control file transfers from command files as an alternative to control from operator's keyboard.
*P/OS version released for DEC Professional*

**1984:** Mechanism built into the package protecting against "message bouncing" due to line noise when computers remain connected and the package is not in use.
*MicroRSX and MicroRSTS versions released for DEC MicroPDP-11.*

**1985:** Commenced development of new portable versions written in the programme language 'C'.

**1986:** First releases of new portable versions written in 'C' for PC-DOS, MS-DOS, UNIX, AIX and VMS

**1987:** Portable versions support simultaneous multiple links
IPL-11 wins ICP Million Dollar Award.

**1988:** PC versions enhanced with improved terminal emulation including VT100 emulation, keyboard mapping and facilities to define function keys.

**1989:** Comprehensive upgrade for unattended operation of multiple PC/host links supporting auto-dialling modems.

**1990:** **Release of MULTI-POLL range of polling software incorporating IPL-11 allowing PCs or other computers at multiple sites to be telephone-polled by a central host computer system to transfer data to and from the central system.**



XOREN COMPUTING LTD
28 MADDOX STREET
LONDON W1R 9PF
TELEPHONE 071-629 5932
FAX 071-629 5432

CIRCLE NO. 129

I am interested in Xoren Comms Software:

Name _____

Position _____

Company _____

Address _____

_____

Postcode _____

EXE 9 91

Presentation Spaces is given in Figure 2. Until a Presentation Space has been connected to the API, you cannot send keystrokes, search for messages, read the OIA or perform any of the other useful functions which the API provides. You will have noticed that the data parameter of this call contains the Presentation Space ID 'E'. When using FTHLLAPI, the PSID of the Host-Connected Presentation Space is always E, as FTTERM only has single-session capabilities.

At this point you have a Host-Connected Presentation Space which is available for use, and may call all of the functions allowing you to control it.

The first thing we will do is to check that the modem is reset to its factory defaults, and that it is responding to our requests. The simplest way to do this is by using the `API_SEND_KEY` function to send the Hayes check status command `'AT&FV1'` to the modem, and then check for an `OK` response using the search presentation space (`API_SEARCH_PS`) function. If nothing is amiss, we send a dial string to it, and wait for a `CONNECT` message to be displayed. This section, in particular, would be a lot more complicated in a real application, as you would need to test for a wide variety of Dial Return Messages.

Once it sees 'CONNECT', the program enters the user and password and waits for the message 'Logged On'. It will then transfer the source file C:\TEST.DAT to the destination DOUGLAS.TEST.TEST, which will be in a predefined storage area on the mainframe. Once this transfer has been successfully completed (the program assumes that it will be), the command X for `exit` is entered, which will log the user off the mainframe. The program is terminated by calling `API_DISCONNECT` (presentation space), `API_DROP_DTR` and `API_UNLOAD` in sequence.

## What's it for?

So, now that we've got a little control over our Mainframe connection, how might we best use it? Well, we could start off by hiding all those messy dial-up and File Transfer progress messages which I was complaining about earlier. There are two HLLAPI session parameters which control screen output - CONPHYS/CONLOG and NOQUIET/QUIET. CONPHYS (CONnect PHYSical) will cause the terminal screen to be displayed, while CONLOG (CONnect LOGical) will not display any terminal activity. This allows you to hide the dial-up process from the user, then display the terminal screen when the user is successfully logged on to the mainframe. NOQUIET/QUIET is the equivalent function pair for file transfer progress messages. With creative use of these functions, and your own progress-reporting routines, you can greatly improve the appearance of the connection and file transfer procedures, while maintaining full terminal session functionality. Note that you could not achieve this with Automator.

Another useful facility is the ability to prevent the operator from typing anything, for example to prevent a critical file transfer from being interrupted. IBM provides two FTHLLAPI functions for this purpose - RESERVE and RELEASE, which lock and unlock the keyboard respectively. One anomaly in FTHLLAPI is that these two functions do not kill the session-switching keys which allow the operator to switch between the Presentation Space, the FTTERM menu and the parent process. Even if you prevent the display of the terminal screen and lock the keyboard, the operator will still be able to switch to the terminal screen. The simple answer is to filter out the session-switching keys using a key-press interrupt service routine, before chaining to FTTERM's key-press ISR. IBM does not recommend this - and neither do I.

What if you need to support interactive mainframe applications, which cannot be automated? When users type some text in an area which is not identified as an input area (very easily done), the keyboard is locked - only the RESET key unlocks it. Inexperienced users typically fail to see the OIA message informing them that the keyboard is locked, and start to hit every key in sight. The solution is for your program to monitor the OIA (using the COPY OIA function). If an error message is displayed, keystrokes should be sent to the session to unlock the keyboard and move the cursor to the nearest input field. It's a good idea to inform the user that the program is doing this, otherwise they will probably panic even more! The WAIT function (4) may also come in useful for this type of error-trapping.

## Conclusion

Hopefully, this article will have whetted your appetite for cleaning up botched terminal emulation comms. The benefits of this approach are great, however the development effort required is not. It is worth remembering that this EHLLAPI standard is being adhered to by almost all new 3270 emulation products, including IBM's OS/2 Extended Edition and the AIX 3270 Host Connection Program for the RS/6000 (HCON), and that the communication technique need not be limited to only Asynchronous Dial-up.

If you have a Small System to Mainframe connection requirement, investigate the possibilities. The chances are that you will be able to provide an attractive and reliable solution. The main limitation on what is possible will be your own imagination.

EXE

```
/*  HLLAPI Template Program - template.c
    Compile with Microsoft C V5.1:

    cl /Ox /AS template.c hllc_s.obj
*/
#define API_RESET       21
#define API_CONNECT     1
#define API_DISCONNECT  2
#define API_RAISE_DTR   81
#define API_DROP_DTR    82
#define API_UNLOAD      84
#define API_SEND_KEY    3
#define API_SEARCH_PS   6
#define API_SEND_FILE   90
#define API_PAUSE       18

#define DRIVE_C         3

main()
{
  int     len,rc;

  system("FTTERM|FTHLLAPI > JUNK.TXT");
  system("DEL JUNK.TXT");

  HLLC(API_RESET," ",&len,&rc);
  HLLC(API_RAISE_DTR,"E",&len,&rc);
  HLLC(API_CONNECT,"E",1,&rc);
  /* Modem Initialisation String */
  HLLC(API_SEND_KEY,"AT&FV1@E",4,&rc);
  /* Pause a second before searching PS */
  HLLC(API_PAUSE," ",2,&rc);

  HLLC(API_SEARCH_PS,"OK",2,&rc);
  /* Dial String */
  HLLC(API_SEND_KEY,"ATD0811234567@E");
  /* Pause a second before searching PS */
  HLLC(API_PAUSE," ",2,&rc);
  HLLC(API_SEARCH_PS,"CONNECT",7,&rc);

HLLC(API_SEND_KEY,"USERID@TPASSWD@E",16,&rc);

  /* Pause a second before searching PS */
  HLLC(API_PAUSE," ",2,&rc);
  HLLC(API_SEARCH_PS,"Logged On",9,&rc);
  HLLC(API_SEND_FILE,
       "TEST.DAT DOUGLAS.TEST.TEST",
        26,DRIVE_C);
  HLLC(API_SEND_KEY,"X@E",3,&rc);
  HLLC(API_DISCONNECT,"E",1,&rc);
  HLLC(API_DROP_DTR,"E",1,&rc);
  HLLC(API_UNLOAD,"E",&len,&rc);
}
```

*Figure 3 - A Simple FTHLLAPI Program*

*Douglas Burns has used C, Clipper, Z80 Assembler and assorted other languages for 8 years. Half this time was spent writing commercial games software, the other half writing business applications. He is currently employed by a major Finance House, with specific responsibility for PC Connectivity applications. He can be contacted on the Solomon BBS (0494) 724946*

*Douglas was using IBM's PC/HOST File Transfer and Terminal Emulation Program (£190). The primary reference manual used was the Personal Communications/3270 EHLLAPI Programming Reference, also available from IBM (£54).*

# The network will crash next Tuesday

*LANs and WANs are increasingly common, but a badly-planned network may cost over $1 million an hour. John Hannawin shows how simulation can help.*

A long, long time ago - say five years back - networking was a rare technology, that had yet to reach the majority of office computer users. At that time, PCs were often viewed as solitary creatures, living a life of lonely splendour on their users' desks. Each computer was the exclusive realm of one person - it would run his favourite programs, and the information on it was for his use as he toiled in glorious isolation.

How times change. Modern computers are no longer isolated beasts; LANs are everywhere and most system developers will need to consider them. It is estimated that more than 40% of business computers are now networked, and a recent IDC survey stated that this was the area where most IT investment was to be concentrated.

This is not surprising - the gains in efficiency that follow from using a network can be immense. However, just as a network of computers contributes more than the sum of its parts, so there are also dangers and problems that appear - as if from nowhere - to disturb the serenity of your life.

These pitfalls can be severe; for many organisations the flow of (computerised) information is now 'mission-critical'. City firms now transfer the bulk of their funds using EDI systems - with hundreds of millions of dollars in transit every second network failures might not be popular! Imagine the problems of an automated factory if a robot is traumatically disconnected from its controlling mainframe. Even in a typical office, using the wrong system could be extravagantly wasteful or cause congestion, delays and annoyance.

***Costing over $1 million per hour, network failures might not be popular***

At a recent 3Com seminar, it was estimated that in an average campus environment an hour of network down-time cost over $10,000. In a financial institution the cost was over $1 million. (Source: Networking Connectivity, Technology & Issues, by Nicholas Lippis). Given those prices, it is probably worth the developers' time to consider how reliable his system will be.

There are a lot of choices to be made, and they are important. And yet, many system developers and programmers find themselves with the responsibility of planning, installing or managing a network with only a limited expertise in network engineering (not surprising, considering how rapidly the field is developing). There are multitudes of products available from a huge range of vendors, running with a variety of different software and hardware. There are different topologies to worry about ('Should I use Ethernet or Token Ring?'); different operating systems; different hardware vendors; different ways of arranging a network; different protocol stacks; choices to be made about using a file-server and so on...

It gets worse.

Network performance is typically stochastic and non-linear. It is impossible to predict with any certainty when a message will be sent, how long it will be, where it will be heading or how many other messages will be in transit too. Consequently, it is usually only possible to predict its performance statistically ('There is a 99% chance that a message packet will arrive within 1 ms'),

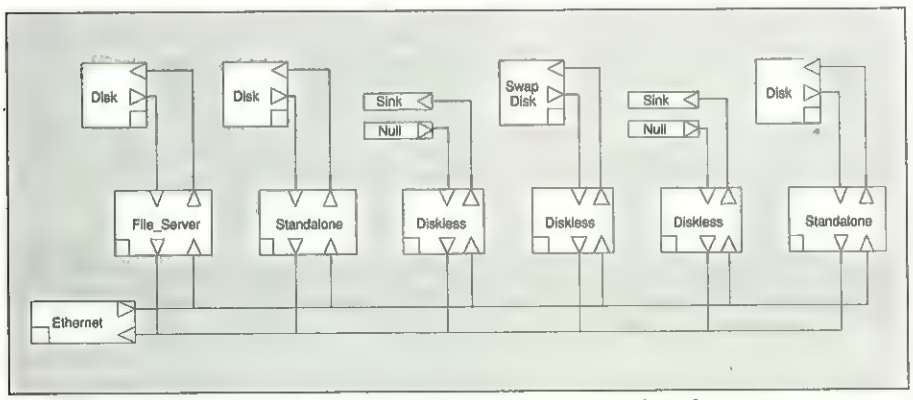



*Figure 1 - A typical model, showing a six node Ethernet spur*

# RECESSION SPECIAL!

PRE-SALE CONSULTING

# TAKE 15% OFF ALL PRICES!

POST SALE SUPPORT

### NEW & NOTEWORTHY

Drover ToolBox for Windows v1.2 ..... £214
A dynamic link library of 200 functions and seven classes for Windows 3.0 development.
Lahey F77L v5.0 ..... £511
New features include allocatable arrays and other important Fortran 90 features.
LANtastic 2Mbps Starter Kit ..... £319
Economic alternative to Novell. Links as few as two PCs and grows upto 100s.
MS CD-ROM Programmer's Library ..... £243
Contains over 70 indexed and cross referenced books and reference manuals, with over 20MB of code that can be included in your programs.
MS Excel v3.0 for Windows ..... £199
Incredible price! Order while supplies last!
MS MS-DOS v5.0 Upgrade ..... £69
Loads into extended memory. Network drivers, TSRs in upper memory. Full screen editor, Undelete, Unformat and more!
MS Visual BASIC ..... £129
Create GUIs in minutes without writing a line of code! Then add functionality with a powerful general-purpose language!
NE-2000 Ethernet Kit ..... £154
16-bit high performance network adapter.
Novell NetWare v2.2 ..... £ 556
Combines Netware 2.x, SFT, Advanced and ELS I & ELS II into one product.
Oracle Tools & Database v6.0 ..... £674
Now includes PL/SQL and SQL*DBA.
Viva 2496if 9600bps FAX/Modem ..... £159
9600bps int. Send/receive fax/modem
Zortech C++ Developer's Edition v3.0 ..... £350
Optimizing C++ Compilers for DOS, Windows & OS/2

### ARTIFICIAL INTELLIGENCE

Anty Combination for Windows ..... £834
PDC Prolog (Turbo Prolog) ..... £201

### ASSEMBLERS & LINKERS

Blinker ..... £201
Dis/Doc Professional ..... £188
Microsoft Macro Assembler ..... £97
Sourcer ..... £105
Spontaneous Assembly ..... £160

### BASIC

Mach 2 for MS QuickBASIC ..... £55
Object Script QuickBASIC ..... £105
PowerBASIC ..... £71

### BUSINESS

ABC Flowcharter ..... £177
Adobe Illustrator for Windows ..... £271
Baler ..... £308
Form Tool Gold ..... £55
Grammatik IV ..... £54
IBM Current ..... £213
IBM Storyboard Live ..... £276
Micrografx Charisma ..... £324
ORG PLUS Advanced ..... £60
Quattro Pro Competitive Upgrade ..... £88
Show Partner F/X ..... £224
SideKick Plus ..... £125
Super Project Plus ..... £238
Xerox FormBase /Windows ..... £297

### C

Aztec C Cross Development Systems ..... £376
Borland C++ v2 0 ..... £295
C Communications Toolkit ..... £129
C-terp 386 with Source ..... £192
CLEAR+ for C ..... £120
Control Pak/Windows ..... £383
DESQview API C Library ..... £110
Essential Communications ..... £194
Greenleaf Comm++ ..... £149
Greenleaf ViewComm ..... £420
Informix C-ISAM for DOS ..... £147
Lattice C Development Sys v6.0 ..... £143
Object Script C ..... £349
PC-Lint ..... £75
Pro-C with Workbench ..... £557
Turbo C TOOLS ..... £96
Turbo C++ Second Edition ..... £63
WATCOM C/386 Professional ..... £904
WindowsMaker ..... £596
XVT (Windows/ PM/ Macintosh) ..... £415
Zortech C++ Database Class Library ..... £193
Zortech C++ for Windows & DOS ..... £196

### CADD

3D Drafting ..... £199
AutoSketch ..... £143
Design Cad 2D ..... £145
Design CAD 3D ..... £193
DRAFIX CAD for Windows ..... £410
DRAFIX CAD ULTRA ..... £237
Generic CADD ..... £281

### CASE

Case: PM for C++ ..... £1,713
Case: Windows Corporate Edition ..... £855
Case: Windows Standard Edition ..... £425
Compeditor II ..... £225
EasyCASE Plus v3.0 ..... £425
Oracle CASE Dictionary ..... £1,504

### COBOL

COBOL spII ..... £415
COBOL/2 Compiler with Toolset ..... £1,449
Dialog System ..... £805
Level II COBOL ..... £398
RM/COBOL-85 Development System ..... £756

### COMMUNICATIONS

Carbon Copy v6.0 ..... £108
Crosstalk for Windows 3.0 ..... £117
LapLink Pro v4.0 ..... £84
Mirror III ..... £81
Procomm Plus ..... £64
Reflection 1 for Windows ..... £289
Remote2 (Host & Caller) ..... £105
SmarTerm 240 ..... £202
WinConnect ..... £58

### DATABASE & FILE MANAGEMENT

Advanced Revelation ..... £705
ALPHA/Four competitive upgrade ..... £91
AskSam ..... £213
Btrieve DOS 3.1 Network for PC ..... £383
Btrieve for Windows ..... £383
Clarion Personal Developer ..... £43
Clarion Professional Developer ..... £444
DataFlex ..... £393
dbFast/PLUS ..... £166
dbFast/Windows ..... £239
FairCom Toolbox Special Edition (1-user) ..... £448
Gupta SQLWindows ..... £910
Informix SQL for DOS ..... £521
Oracle Tools & Database v6.0 ..... £670
Paradox v.3.5 ..... £488
Paradox SQL Link ..... £335
R&R Clipper and FoxBASE+ Module ..... £42
R&R Relational Report Writer for dBASE ..... £160
R&R Report Writer for Oracle SQL ..... £192
R&R Worksheet Report Writer ..... £96
R:BASE Single User ..... £469
SQLWindows for Btrieve ..... £1,370
Superbase 4 Windows ..... £418
UI Programmer ..... £320
XQL Relational Data Manager ..... £529
Xtrieve Plus ..... £403

### DEBUGGERS

ClearView-386/486 Debugger ..... £318
Multiscope Debugger for DOS ..... £114
Multiscope Debugger for Windows ..... £240
Periscope II with Breakout Switch v5 0 ..... £121
Turbo Debugger & Tools v2.0 ..... £97

### DESKTOP PUBLISHING

Adobe PostScript Cartridge ..... £297
Adobe Type Align /Windows ..... £59
Adobe Type Manager for Windows ..... £60
CatchWord OCR ..... £136
ColoRIX VGA Paint ..... £97
Facelift for Windows ..... £58
Fontware Installation Program ..... £56
Fontware Typeface Packages ..... £77
Go Script Plus ..... £179
Inset + Hijaak ..... £160
OmniPage Pro ..... £598
OmniProof for DOS ..... £99
PageMaker ..... £493
Persuasion for Windows ..... £303
The Complete Page Scanner w/ OCR ..... £579
WordScan 286/386 2 Mb ..... £383
Xerox Ventura Publisher Gold v3.0 ..... £504

### DISK/DOS/BACKUP

Back-It ..... £74
Copy II PC ..... £26
Disk Manager ..... £35
Norton Commander ..... £94
Norton Utilities ..... £110
OPTune ..... £43
PC Tools with AntiVirus ..... £97
Q-DOS II ..... £43
SpeedStor Universal ..... £61
SpinRite II ..... £58
WonderPlus ..... £47
XTree Pro ..... £54
XTree Pro Gold ..... £89

### EDITORS

Brief v3.1 ..... £176
EDT + ..... £213
KEDIT ..... £97
Norton Editor ..... £47
Sage Professional ..... £189
VEDIT PLUS ..... £110

### FORTRAN

Fortran Toolkit 2 ..... £116
GRAFMATIC & PLOTMATIC ..... £155
RM/FORTRAN Development System ..... £345
Spindrift Utility Library (F77L) ..... £79
ULI Maths Functions Library (F77L) ..... £188

### GRAPHICS

Corel Draw ..... £374
Designer for Windows ..... £455
Dodot ..... £111
Draw Applause ..... £263
GRASP with ARTOOLS ..... £191
GSS GKS Kernal System for DOS ..... £415
GSS Graphic Devlpnt Toolkit for DOS ..... £555
HALO ..... £227
HALO Professional ..... £341
Halo Windows Toolkit ..... £338
Harvard Graphics ..... £301
HiJaak ..... £128
MetaWINDOW ..... £222
PCPaintBrush for Windows ..... £52
Pizazz Plus ..... £85
Windows Draw Plus ..... £160

### HARDWARE

8087-10 Math Coprocessor ..... £157
ASCII Internal 2400 Modem ..... £68
ASCII Internal 9600bps V.32 with MNP 5 ..... £428
FastMath 83S87-16 Math coprocesor ..... £243
Hardcard IIXL 105Mb, 9ms Hard Disk ..... £555
Logitech TrackMan Serial Mouse ..... £76
MouseMan Serial Mouse ..... £68
Pacific Page PS Cartridge for HP IIP ..... £348
Paradise 1024x768 VGA w/ 512k ..... £191
Practical 2400 Pocket Modem ..... £106
ProScan PS1000 64 Grayscale Scanner ..... £520
SatisFAXtion Classic ..... £375
ScanMan Plus PC ..... £171
The Complete Communicator ..... £412
The Complete Page Scanner ..... £530
VIVA 9600 V42 bis External Modem ..... £446
Weitek 3167/25MHz Math Coprocessor ..... £558
Worldport 2496 Portable Fax ..... £465

### INTEGRATED

Framework III ..... £424
Lotus Symphony ..... £485
Lotus Works ..... £89
Microsoft PC Works ..... £92

### LAN

LANtastic AE-2 Ethernet Starter Kit ..... £450
NE-1000 Ethernet Kit ..... £125
Netware v. 3.11 20 user ..... £2,327
Netware C Interface for DOS ..... £196
Oracle for Server ..... £1,254
Procomm Plus for Network ..... £325
Q-DOS II for Network (8-user) ..... £134
Saber File Manager ..... £200
Saber LAN Administration Pack ..... £351
Saber Menu for MS Windows 3.0 ..... £249
Saber Menu System Single-User ..... £45
Saber Meter Usage Tracking Software ..... £99
Saber Secure ..... £75
SoftwareBridge LAN ..... £143
Super Project Expert Lanpak ..... £420
Superbase 4 Windows Network ..... £598

### MAC

Adobe Illustrator Mac ..... £332
Adobe Type Manager ..... £56
Carbon Copy Mac ..... £108
LapLink Mac III ..... £84
MathCad ..... £305
Mathematica ..... £425
Norton Utilities for Mac ..... £82
Oracle Developers Kit Mac ..... £501
Pagemaker ..... £486
PC-Matlab /Mac ..... £752
Persuasion ..... £311
Quark Xpress ..... £512
Smalltalk/V for Mac ..... £107
Zortech C++ for MAC ..... £196

### MEMORY MANAGEMENT

386 Max ..... £60
Above Disk ..... £58
Headroom ..... £70
QEMM 386 ..... £57
Turbo EMS ..... £48
Vcache ..... £45

### MODULA-2

EmsStorage with Full Source ..... £48
Logitech Modula-2 Development System ..... £158
Logitech Modula-2 Window Pack ..... £30
ModBase ..... £21

### OBJECT ORIENTED PROGRAMMING

C++ Views ..... £319
Codebase ++ ..... £222
HCR/C++ ..... £727
Smalltalk/V ..... £69
Smalltalk/V 286 ..... £107
Smalltalk/V Communications ..... £27
Smalltalk/V EGA/VGA Colour Pack ..... £27
Smalltalk/V Goodies 1 ..... £27
Smalltalk/V-Windows ..... £268
Zinc Interface Library ..... £160
Zortech C++ MS-DOS 386 ..... £641
Zortech Windows Toolkit ..... £45

### OPERATING ENVIRONMENTS

386/ix Operating System Release 3.2 ..... £330
DESQview ..... £77
DR DOS ..... £130
DR Multiuser DOS ..... £448
GeoWorks Ensemble ..... £105
Microsoft Windows ..... £92
OS/2 Extended Operating System ..... £662
OS/2 Standard Operating System ..... £258
SCO UNIX System V/386 v.3.2.2 ..... £370

### OS/2

Corel Draw for OS/2 ..... £320
Describe Word Publisher (Client) ..... £164
Display Write 5/2 ..... £234
Halo for OS/2 ..... £338
Microsoft OS/2 Presentation Mgr.Softset ..... £92
Microsoft OS/2 Presentation Mgr.Toolkit ..... £308
Oracle OS/2 ..... £652
PageMaker for OS/2 ..... £486
Paradox OS/2 ..... £462
Qedit for OS/2 ..... £48
QEMM Expanded Memory Manager 50/60 ..... £58
Smalltalk/V-PM ..... £268
Super Project Expert/2 ..... £481
ToolBook (OS/2) ..... £280
TopSpeed C++ for OS/2 Standard ..... £213
Vitamin C for OS/2 ..... £319
Wide Angle for OS/2 ..... £62

### OTHER

ACT! (Sales lead tracking) ..... £233
Adlib Card (Sound) ..... £104
Builder (Batch Compiler) ..... £81
graphics-MENU (GUI) ..... £134
Guide 3 for PC (HyperText) ..... £358
HP NewWave (GUI) ..... £105
HyperPAD ..... £82
Install (Automatic installation) ..... £148
KnowledgePro ..... £420
Menuet (GUI Development) ..... £209
PolyAWK v1.3 ..... £126
RPG II Development System ..... £1,117
SCHEMA-PCB (SC-4) ..... £733
SoundBlaster ..... £147
System Sleuth (Diagnostics) ..... £72
VirusCure Plus ..... £64

### PASCAL

Asynch Plus ..... £121
B-Tree Filer ..... £97
Graf/Drive Plus Personal ..... £81
Object Professional ..... £132
Pascal TOOLS ..... £81
POWER TOOLS PLUS ..... £96
Topaz ..... £64
Turbo Analyst ..... £90
Turbo Pascal for Windows ..... £150
Turbo Pascal Professional ..... £190
Turbo Professional ..... £90

### SCIENCE & ENGINEERING

ChiWriter Chemistry Support ..... £45
Derive v2.0 ..... £183
InGraf ..... £263
Labtech Notebook ..... £801
Matfor - 286 ..... £225
MathCad for Windows ..... £303
Mathematica 386 ..... £597
Mathematica for Windows ..... £855
PlotLib ..... £188
SSP/PC Scientific Subroutine Library ..... £245
Tech*Graph*Pad ..... £335
Zortech C++ Science & Engineering ..... £490

### SCREEN DESIGN & MANAGEMENT

C Windows Toolkit ..... £86
Flash-Up ..... £69
Hi Screen XL ..... £112
POWER SCREEN ..... £96
ProPak ..... £412
SayWhat!? ..... £33
Vermont Views ..... £319
Vitamin C ..... £254
WNDX GUI Development System ..... £321

### TRANSLATORS

BAS_C Commercial ..... £242
FOR_C ++ ..... £870
FOR_C with Runtime Source ..... £628
Pascal to C Translator ..... £140
Turbo Plus ..... £138

### UNIX/XENIX

Architech Series Workstation Developer ..... £1,528
DataFlex for UNIX 386 ..... £950
Informix C-ISAM (SCO 4.0) ..... £295
LPI COBOL Compiler ..... £1,076
Norton Utilities for System V v3.2 ..... £174
Oracle UNIX/Xenix 386 ..... £1,476
SCO UNIX System V/386 Dev.System ..... £641
VEDIT PLUS for UNIX/XENIX ..... £184

### WINDOWS DEVELOPMENT

Asymetrix ToolBook ..... £280
BeckerTools ..... £73
Microsoft C with Windows SDK ..... £466
MS Windows Device Driver Dvlpnt Kit ..... £311
Microsoft Windows Productivity Pack ..... £38
Object Vision ..... £86
WIN++ ..... £160
WindowsMaker Professional ..... £746
WinSleuth ..... £72

## USA SOFTWARE

2 OLD BROMPTON ROAD • SUITE 315 • LONDON SW7 3DQ
PHONE: 071–351 2360 • FAX: 071–352 0974

**Over 3,500 products carried. Call today for price list-disk.**

## We will match any advertised Price!

- Prices exclusive of V.A.T.
- All major credit cards accepted.
- 30 day invoicing for qualified organizations.
- Special discounts for large or corporate buyers.
- Prices subject to change without notice.

UK8/91

CIRCLE NO. 130

and it is difficult to extrapolate from known cases; a system with 200 nodes will behave very differently to four 50 node systems. Some protocols are inherently unpredictable (eg Ethernet's famous binary exponential back-off algorithm). Calculating figures from scratch is difficult (due to the huge amounts of statistics required); however it is better than the approaches sometimes adopted.

It would obviously be impractical (not to mention very expensive) to assess the suitability of a network by actually building it and seeing how it behaves! Similarly, users may not appreciate rehearsing a network crash for real, 'just to see what happens'. As a result, despite their importance, most networks are planned, designed and managed on the basis of little more than informed guesswork, common-knowledge or a touching faith in the patter of a sales rep.

One technique often suggested is to use a network analyser (eg Data General's Sniffer). However, while these can help to monitor the performance of an existing installation and are excellent as diagnostic tools, they are less valuable in planning a new system. They can't predict how an existing network's performance will change (for instance, as load, equipment or topology vary), nor can they evaluate the performance of a network that has yet to be built.

Simulation, however, can be used for just these roles. Just as a spreadsheet allows the financial manager to ask 'What if' questions about money, a network simulator allows system managers to investigate 'What ifs' with their LANs and WANs. They can thus be used to evaluate the performance of existing networks and also to predict changes as the system is modified. Simulations can also be used for evaluating the comparative performance of different designs, protocols or vendors.

During the past two decades, a number of approaches have been developed for modelling and simulating communication networks. While all of these approaches use an event driven simulation technique, they differ in the methods used to construct the network model and the simulation programs differ widely. Early approaches to network modelling and simulation relied on programming languages such as GPSS, SLAM, and SIMSCRIPT or even - horror of horrors - C. This may seem a sensible technique to .EXE readers, expert programmers all. It may seem less reasonable when you consider the enormous complexity of most network protocols and the amount of time and programming effort required to develop such a model. It would be an unusual accountant who chose to write a spreadsheet from scratch; why should the person paid to be a network manager be expected to give up his day job and become a part-time SIMSCRIPT programmer?

***Network performance is typically stochastic and non-linear - it is impossible to predict with any certainty***

## Enter the Specialist

Instead it makes more sense to use a specialist program. Typically, these use abstract (mathematical) paradigms which are then translated (either manually or automatically) into simulation programs. Graph-based network models, such as finite state machines and Petri nets, are typically used for protocol specification and validation. However, in real world situations, these produce overly detailed models, unsuitable for simulating the performance of a large network. An alternative is the use of an extended queuing paradigm, which models a network as a set of resources (for instance links, buffers and controllers), and a corresponding set of jobs (for example polling, error recovery, acknowledgement) which require those resources. However, this is inappropriate for analysing protocols, and can be inflexible or hard to modify.

There are now several programs which have been developed to help tackle these tasks. Typically, these programs provide libraries of the most common protocols and functions (PCs, file-servers, bridges etc) which can build a representation of the system and provide an estimate of how it will perform - before shelling out any hard cash on the real thing!

In this article, you must forgive me for concentrating on our own product: Comdisco Systems' Block Oriented Network Simulator (BONeS). However, many of the points discussed apply to any professional simulation package.

## How to model a system

A network is built up of nodes, joined by communication links (or arcs). In all cases, packets of information are then passed from function block to function block, being operated upon as they travel. The user must define the function blocks and the structure of the data packets as the first step in modelling a system. The simulator operates on an event-driven basis: 'WHEN a packet arrives at a block THEN do ...', and handles all details.

The first step in modelling is to map the system topology; what parts are involved and how are they connected. At the lowest level the function blocks are called primitives and perform simple operations, such as delaying a data packet for a fixed (or programmable) time or modifying a field within the data structure.

A number of more complex models for network elements and data structures are usually supplied too; including traffic sources, arithmetic and logical operators, channels, timers, delays and other elements needed to construct protocol function. Using the primitives and the building blocks from the library, the user can construct a model of a network as shown in Figure 1. When comparing simulators it is important to check what is included in their libraries; a good package will support all

How easy to use is it? (Graphical systems can be 2-5 times faster in use than text-based ones.)

How many models does it include?

What protocols does it support?

Is it an 'Open' system? Can you develop new models or are you restricted to existing ones?

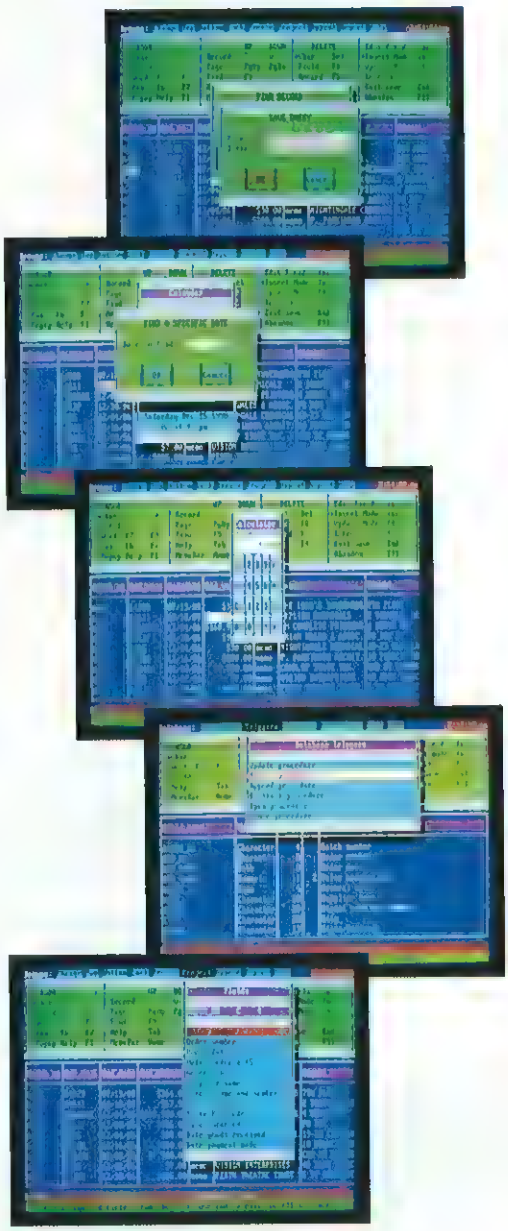
Does it support model hierarchy or abstraction?

How accurate are the results?

*Figure 2 - Considerations when choosing a simulator*


DOS POWER TOOLS
TECHNIQUES, TRICKS AND UTILITIES
PAUL SOMERSON
REVISED FOR DOS 5.0
2ND EDITION
OVER 100 BRAND NEW UTILITIES ON THREE DISKS
The all-time worldwide bestseller, now revised and updated for DOS 5.0
'A fabulous work, in a league by itself... a must buy' John Dvorak
THE BEST JUST GOT BETTER.
BANTAM COMPUTER BOOKS
THE POWERHOUSE BEHIND POWER TOOLS
1-2-3 POWER TOOLS
WINDOWS 3 POWER TOOLS
NOVELL NETWARE POWER TOOLS
All available now from your local computer bookseller, or order direct on our 24-hour credit card hotline 0326 77781
For further information, please contact Stephen Page, Sales and Marketing Manager, Bantam Computer Books, Transworld Publishers Ltd, 61-63 Uxbridge Road, London W5 5SA
Bantam Computer Books is a Division of Transworld Publishers Ltd.
CIRCLE NO. 107

# Recital 7.0 Is The Only Choice.



- Advanced RDBMS/4GL
- Active Data Dictionary
- Object-oriented with pre-defined "application objects"
- Open client/server architecture
- Integrated pop-ups for calendar, calculator, choice lists
- Built in "Dynamic Compiler" New v7.0
- Pop-up time management scheduler New v7.0
- Relational multi-record table fields in forms New v7.0
- Split screen relational browse New v7.0
- Database, form and report triggers New v7.0
- Automatic record locking
- Supports 14 datatypes including memo, numeric, byte, integer, short, character, date, logical, float, real, packed, zoned, vaxdate, quadword
- Available now on 100+ VAX/VMS and UNIX computers (including full VAX cluster support)
- SQL (interactive and embedded)
- dBASE, FoxBase and Clipper compatibility (yes, even on VMS!)
- Terminal (including full color support) and keyboard independence
- Transparent connectivity to existing RMS and RDB databases on VAX/VMS
- Completely integrated development environment
- Powerful end-user tools

Recital CORPORATION

RdbSVP SOLUTIONS · VENDOR · PROGRAM

88open

Recital Corporation LTD, South Bank Technopark, 90 London Road, London SE1 6LN, United Kingdom, Tel 071-401-2727, Fax 071-633-9617
Recital Corporation Inc., 85 Constitution Lane, Danvers, MA 01923, Tel (508) 750-1066, Fax (508) 750-8097

CIRCLE NO. 132

the standard units you may need (file-servers, bridges, routers, disk-less workstations etc).

BONeS is totally graphical - systems are drawn on the screen just as in a 'back of the envelope' sketch, by selecting icons from the libraries and linking them together. A graphical simulation language is very much easier to use and understand than a text-based system. In addition, the system can be regarded as being self-documenting; the diagram shown in Figure 1 not only defines the system but can be readily understood.

As everyone who's learnt the Gospel according to Djikstrata knows, systems should be structured and hierarchical - and a good simulator will be no exception. Primitives, existing models and sub-units are used to build up a new item, which can then be saved. From then on it can be treated as a single, sealed unit, or can be 'zoomed' and examined in detail. It is thus possible to develop a block diagram where London communicates with New York, Paris and Chicago; each built up from blocks representing separate sites, which have further sub-units representing individual computers, which in turn contain discrete blocks and primitives.

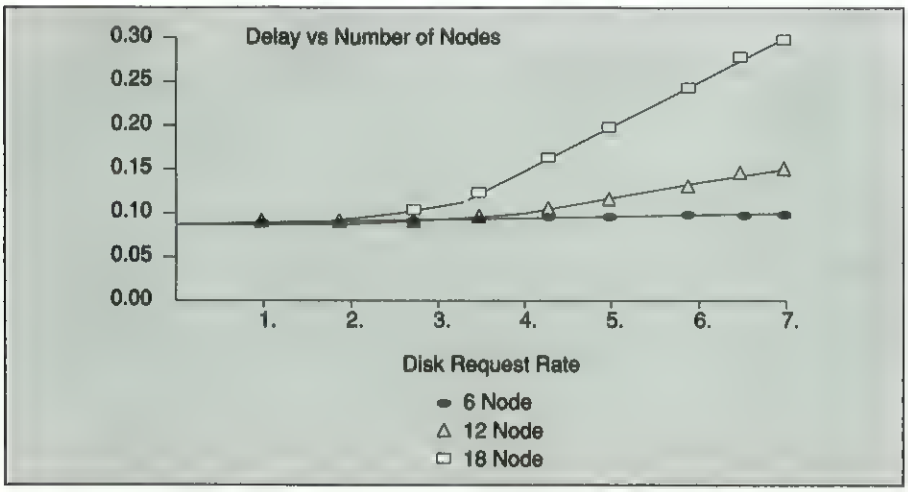



*Figure 3 - Simulation results for a LAN*

When dealing with larger models this can be both complex and unnecessary; an alternative approach is to abstract the information needed. For example, a file-server may be constructed as a data source, a delay, and an Ethernet node. Once these have been defined, it is only necessary to refer to the block as, say, a Compaq Systempro model XYZ.

Some simpler systems only support 'flat' models, with no hierarchy or levels of nesting allowed. These should be avoided for all but the most trivial of applications.

Defining the models and network topology is only half the story; the data packets involved must also be defined. In BONeS these are also created graphically, using the data structure editor, DSE. The data structure specifies the information that will be passed around the network; it's usually the

basic network packet with additional fields added by the model designer to document the flow of the packet through the model. These structures can have an arbitrary number of fields and are also hierarchical; each field can contain simple entities, such as an integer representing the packet length, or more complex entities, such as another complete structure.

Once more, you should check which data structures, both standard and primitive, are supplied within the library. (BONeS supports ALOHA (plain & slotted), CAN, CSMA/CD, FDDI, HDLC/SDLC, LLC, MAN, Sun NFS, SCSI, Token Ring and X.25). It is worth stressing the value of an 'Open' system; one where models and protocols can be accessed, modified or extended by the user. This is essential when investigating new protocols, developing or modifying existing ones. Some computer scientists have exploited this aspect of BONeS to investigate the behaviour of different data structures *per se*, without a network in sight!

Once the definition of the network model is complete, BONeS performs a variety of error and consistency checks. It then comes to the heart of the task and, in essence, 'compiles' the graphical model of the network to produce a C program. This is then executed to generate an event driven *Monte Carlo* simulation of the model, with user specified values for model parameters inserted as appropriate.

The data collected during the simulation are analysed and displayed graphically. Performance measures such as delay, throughput, buffer size and utilisation can be easily computed and displayed.

The results of such a simulation can be used to evaluate different decisions in the development of a communications system (What will be the throughput if we use this packet size? Will performance improve if we add another link? Where is the bottle-neck in this system?). Similarly, a network manager can use these simulations to study the effects of changing the mix of applications, adding faster disks or expanding the network (What will happen if we add another 10 users? What will be the effect of connect-

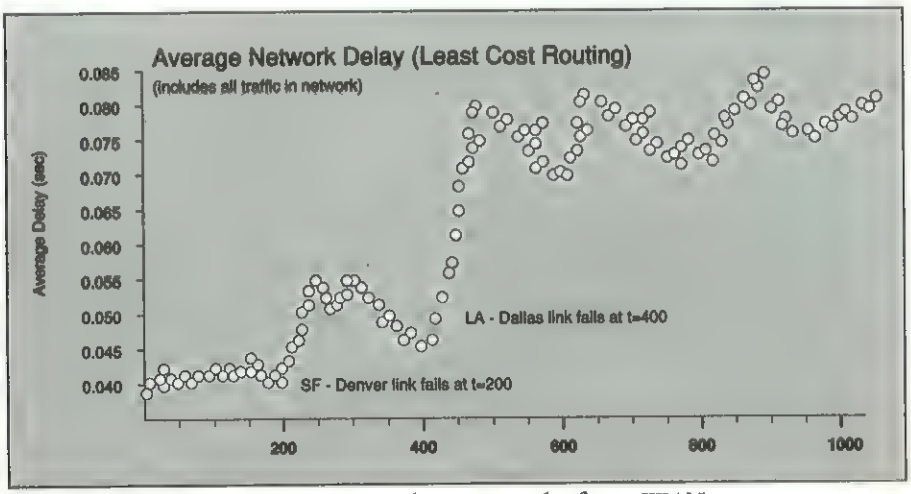



*Figure 4 - Simulation results for a WAN*

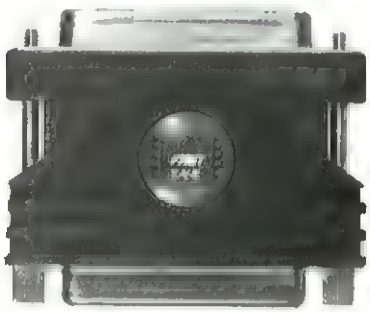
# 5 out of 5 hackers prefer other software protection methods to Hardlock E-Y-E®

**What hackers dislike...**
Hardlock E-Y-E was designed using cryptographic principles. It took the experience and know-how of Germany's No. 1 in software protection and the leading edge technology of a US semiconductor company to create the ultimate software protection tool. Hardlock E-Y-E is based on a custom chip featuring secure algorithmic response rather than simple bit swapping or counting schemes.

**What software developers like...**
Hardlock E-Y-E combines all the features software developers require in a single product: algorithmic response to provide security and an optional non-volatile memory to allow custom configurations. FAST Electronic has made implementation of Hardlock E-Y-E in your software easy. Use HL-Crypt to protect .EXE or .COM files, or incorporate high level language interface routines in your software. The algorithm parameters and the contents of the memory can be programmed in seconds using our Crypto-Programmer card. This unique card guarantees that no one else can burn your original codes. Simply plug the card into any PC slot and start up your own Hardlock E-Y-E workshop.

**What your customers will like...**
Hardlock E-Y-E allows unlimited backup copies. The device is shipped with the software for the user simply to plug into the parallel interface and forget. Daisy chainability, outstanding reliability (no battery is needed), and the most compact High-Tech design ensure that your customer will accept Hardlock E-Y-E.

**What your accountant will like...**
Hardlock E-Y-E needs no factory coding. This ensures optimum delivery schedules and stock flexibility. Revenues will go up as software piracy and multiple usage are prevented. Despite its wealth of features, Hardlock E-Y-E's prices remain competitive.



**Hardlock E-Y-E programmable, algorithmic response and memory option – all in one.**

...As more and more software developers, customers and accountants appreciate the Hardlock E-Y-E device, hackers like it less and less.

MAGNIFEYE

**Order your demo unit today. Contact Magnifeye,**
**235-239 Walmer Road, Walmer Studio #6, W11 4EY, Telephone 071 221 8024, Fax 071 792 3449.**

CIRCLE NO. 135

**NEW VERSION 3**

***Extra features include:***

Plotter Support ■ WBS & OBS Coding ■ High-Resolution Charts ■ Proposal Format ■ Extended Printing Features ■ End Date Scheduling ■ Extended Importing & Exporting ■ Optional LAN Support

# Put InstaPlan to the test for 30 days with no risk

*Satisfaction guaranteed – or your money back!*


Whatever your project planning or management needs, you can't make a better choice than the InstaPlan award-winning family of planning software.

Instaplan is widely recognised as offering the best combination of top performance with low price.

From as little as £149 you have everything you need to analyse projects, schedule activities, allocate resources and make all those important deadlines.

InstaPlan is supremely easy to learn and use ... yet offers multi-project support, integrated work accounting, trend analysis and many other advanced features.

Why not put InstaPlan to the test for yourself? You've nothing to lose – if you're not entirely satisfied with InstaPlan, for any reason, simply return it within 30 days and we guarantee to refund your money immediately and in full.

For more details, clip and return the coupon below. Or call us now on 081-908 2423.

## InstaPlan

Deepak Sareen Associates,
28 Preston Road, Wembley HA9 8BR.
Tel: 081-908 2423. Fax: 081-904 6668.

### *Read what the experts say:*

*"Highly professional, highly competent, tremendous range of planning choice, terrific value-for-money."*
**GOOD SOFTWARE GUIDE**

*"InstaPlan is proof that simplicity of design and operation can be achieved without abandoning a good set of project management features."*
**PC MAGAZINE**

*". . . Useful . . . intuitive . . . practical . . . well worth the price."*
**PROJECT MANAGER TODAY**

*"InstaPlan offers the features of a £2,000 package with none of the trade-offs of its low-cost competitors."*
**PERSONAL COMPUTER MAGAZINE**

*"Technically it is extremely competent, being one of the only programs that is capable of levelling resources across multiple projects."*
**COMPUTER SHOPPER**

*"So simple to use that you wonder why other packages make a meal of resource allocation."*
**PC MAGAZINE**

*"Excellent rethink of requirements for project management packages . . . Ease of use – excellent, functionality and documentation – good."*
**PC USER**

BEST PC MAGAZINE BUY

PERSONAL COMPUTER EDITOR'S CHOICE

WHICH COMPUTER? BEST BUY

**UPGRADE FOR JUST £85!**

*Already using planning software? Now you can upgrade to top-of-the-range InstaPlan EMS for just £85 – a saving of £460 on the normal selling price! Send today for full details.*

Please send me a free information pack on InstaPlan, and details of the special InstaPlan EMS upgrade offer. Plus how to recrive my **FREE** guide to 'Getting the job done' (offer closes 30th September 1991).

Business Computing 91

**FREE TICKET** simply for replying

Name _____

Job Title _____

Company Name _____

Address _____

_____

Postcode _____ Tel _____

Do you currently use any project management software?

☐ NO ☐ YES If YES, please state which

*POST TODAY – NO STAMP NEEDED*

**DEEPAK SAREEN ASSOCIATES**
FREEPOST, 28 Preston Road, Wembley HA9 8BR

Tel: 081-908 2423 Fax: 081-904 6668

EXE 9/91

CIRCLE NO. 136

ing these two sites? How would performance improve if we used a file-server?)

A simulation is only of value if its predictions can be relied on. A misleading forecast is worse than no forecast at all. It is hard to be exact, as much depends on how well the model is developed, but a difference of 1% or less between forecast and actual (tested) results is not uncommon with a good model and simulator.

By using a powerful program, developers can focus their attention on designing and analysing the system - rather than on writing and debugging simulation programs.

A modern simulation system (such as BONeS) will help network developers and specifiers to get things right, saving themselves and other users a lot of worry and potentially expensive failures.

## Two Examples

Figure 1 shows a typical LAN, with six computers (including a file-server, stand-alone and disk-less workstations) connected to an Ethernet spur. For this model, the user can specify values for parameters such as network size, traffic rates, transmission rate, disk size and access time.

Figure 3 shows the results of the BONeS simulation for three different network sizes (6, 12 or 18 nodes). Results such as these can be used by a network manager to predict the performance of the network due to projected growth. The network manager can also use these simulations to study the effects of adding larger and/or faster file servers. Note how the three networks behave similarly for low loads, but increasingly diverge as traffic intensifies.

Figure 4 shows how BONeS can be used to model an X25 based WAN (wide area network) linking eleven large computer sites in cities across the US (Seattle, Boston, New York etc). The model is used to evaluate the consequences of a link failure, calculating how the average delay and utilisation will vary as messages are rerouted.

These transients in a network often have crippling effects which do not normally show up in average statistical measures. For example, as one link fails there will be a rapid loading on to adjacent nodes, as messages are redirected. This can overload further stations which fail in turn, until a catastrophic 'Domino effect' results (the infamous AT&T network crash of the 1980s is the best known example, but there have been many more). Obviously, this information is very valuable; without simulation it is virtually impossible to determine.

System developers can use this information to assess their designs and compare different possibilities in the search for the best configuration. In this example, the trend of steadily rising delay and utilisation may cause concern.

EXE

*John Hannawin is European Sales Manager of Comdisco Systems. Before joining Comdisco, he spent several years with Hewlett Packard in Bristol, working with both the sales team and the Network Research Laboratory.*

*BONeS V1.5.1 runs on workstations from Sun and DEC, and is available from Comdisco Systems on 0454 614 256. It costs $25,000 for single-user license, and $30,000 for multi-user license (pan-European prices).*

# TopSpeed Modula-2 V3.0 - A top gear release?

*JPI started shipping Version 3.01 of the latest TopSpeed program development products earlier this year. Richard Pickard has had a go at driving some of them.*

The TopSpeed product is a complete environment for DOS or OS/2 program development. It has its own multi-windowing editor with context sensitive language help, compilers, project system, smart linker and debugger. The environment is highly configurable and the project system can cope with make-like scripts of considerable complexity.

Jensen and Partners International's specialisation is languages. Although its language libraries are comprehensive, and there is a raft of utility programs in the optional TechKit, it does not provide application-oriented utilities (eg a screen designer) or its own OOP class libraries.

## Installation

Let's plunge right in. With enough space on the hard drive, it was only an 11 minute job to do installation and even simpler than with earlier releases. At the end of the procedure it does a little tidying up of its own files. As before, it doesn't tidy up your files; you change your own AUTOEXEC.BAT and CONFIG.SYS. I installed the environment - the foundation for all languages - Modula-2 with source kit, and the TechKit. These took up 6 MB, twice as much as in V2.0, but I did include extra material for mixed language work and Windows and four memory models instead of three.

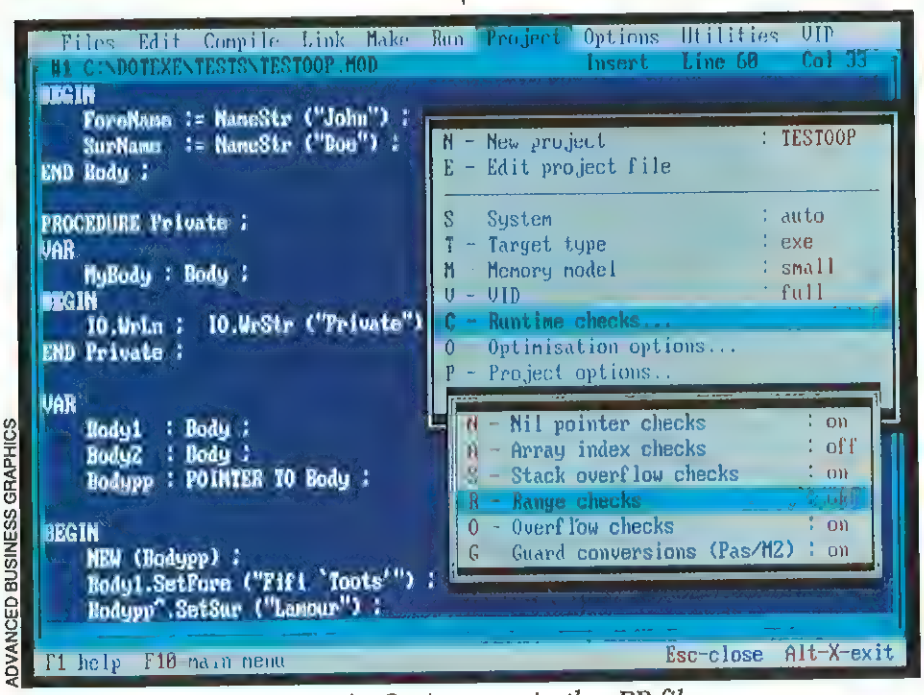



*Figure 1 - Options are in the .PR file*

The TopSpeed languages available are Modula-2, Pascal, C and C++; all compilers share the same code generator. I've used the earlier releases of Modula-2 and I'll be concentrating on that. *(The other components in JPI's new TopSpeed range - the C, C++ and Pascal compilers - will be covered in a separate article next month - Ed.)*

## The Environment

The TopSpeed editor/environment is now purchased as a separate package from the languages, and one copy suffices for all languages. When you start up the environment to see how it handles, the visual differences are most obvious when you're not editing source files. There is a new welcome banner and, instead of a popped up menu in an otherwise empty workspace, there is a permanent top-line menu. All the menus' default colours are changed (for the better, less garish) and the information displays are more comprehensive. For example: during a Make, the reason for each compilation is shown alongside the module name (as: pragma changed, object missing, source changed etc).

The handling is unchanged from V2, except for a few additions. As before, the environment is configurable - altering and adding to menus using the utility supplied, and making up keyboard macros directly in the environment - and there are the same shortcuts and utilities. There is some forward compatibility of session files from V2 to V3: old session files are interpreted correctly in respect of edit-buffer contents, but not in respect of colours and sizes. That little 'tidying up' done by the INSTALL program is most obvious when you create a new pro-

# WDA Conference

## 25th September 1991 - Central London

**Following the success of the first WDA Conference in March, the Windows Development Association will be presenting two half day seminars on Wednesday, 25th September 1991 at the Church House Conference Centre in Central London. Using a combination of lecture material and demonstrations, delegates will share the experiences of experts in the Windows arena both from the UK and Europe.**

### MORNING SESSION

**Windows Corporate Strategy**

- THE BUSINESS VALUE OF WINDOWS
  John Scott-Simpson, Shell International, The Netherlands
- WINDOWS IN THE CORPORATE ENVIRONMENT
  Danny Dawes, Credit Lyonnaise
- WINDOWS SYSTEMS STRATEGY
  Bill Champion, Microsoft
- PANEL SESSION

### AFTERNOON SESSION

**Windows Development Techniques**

- CLIENT-SERVER COMPUTING
  Steve Baker, QA Training
- WINDOWS USER INTERFACE DESIGN
  Mike Butler, MSS Systems Ltd
- TECHNICAL CONSIDERATIONS OF IMPLEMENTING MULTIPLE DOS ENVIRONMENTS
  Tim Snape, Abbotsbury Software
- PANEL SESSION

The fee for each session is £59 + VAT (£69.33) for members of the WDA and £79 + VAT (£92.83) for non-members. Delegates wishing to attend both sessions will be entitled to a reduced entrance fee and buffet lunch with prices at £108 + VAT (£126.90) for members and £148 + VAT (£173.90) for non-members.

PRE-PAYMENT IS ESSENTIAL - ENROLMENT LIMITED - PRIORITY TO MEMBERS

### CONFERENCE BOOKING FORM - 25TH SEPTEMBER 1991

Name: _____

Company: _____

Address: _____

_____

_____

Tel No: _____

Fax No: _____

☐ Please enrol me as a member of the WDA @ £95+VAT

Please book a place on:

☐ Morning Session ☐ Afternoon Session ☐ Both

Cheque enclosed for £ _____

☐ Access ☐ Visa

Expiry ☐☐☐☐

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Signature: _____

Credit card address (if different)

_____

_____

.EXE 9/91

All bookings must be received by 20th September and are subject to availability.

Please return to:-

Windows Development Association, Cecily Hill Castle, Cirencester, Gloucestershire, GL7 2EF, England

Tel: (Intl +44) 0285 655888 Fax: (Intl +44) 0285 640181

CIRCLE NO. 138

ject file and pick the memory model; only the models you've installed are displayed as options.

A strategic change from V2 to V3 is to store all the compilation and linking options in a single file for each project and to set them using a single menu. (Figure 1). (In the earlier versions, some of the options were stored in the environment.) Instead of using the Options (Compiler), Options (Make) and Project menus for three sets of controls, they are now combined. This is a genuine simplification for the casual user and a speed-up for the practised.

The options which can be set into the project file include optimisations and run-time checks. The latter will pick up subscript and range errors, memory and stack overflow and the dereferencing of `NIL` pointers. It is a hot argument in many development teams whether or not to leave such checks in place, when a program is supposedly fully tested. My preference is to leave them so that I am forced to correct late-arriving errors or to detect and handle them explicitly. One of my applications does huge amounts of `SHORTINT` arithmetic and array subscripting. During its development the run-time checks were invaluable but I broke my own rule when I discovered that the program ran twice as fast if I took the checks out. It is as well to be aware of the cost of the checks.

In appearance, the environment is not quite the same as a 'desk-top', but it has the same value for a program developer. If the end-product is a program, you can run it or debug it from within the environment. If a run-time check ends a test then you can be taken directly to the source line where it occurred. With the Visual Interactive Debugger (VID) there is full animation of the source, control over breakpoints, watching variables and examining and changing values. These features are in direct descent from earlier versions.

A particular VID headache occurs when a run stops with a fatal error. Sometimes the very fact of an error occurring gives enough information to sort out a problem. If it isn't, you have to restart the run and set breaks 'just before' the point where the error arises; this isn't always easy. The program information you really want isn't available after a crash; you can't get at the call stack or at current values which might help.

## The Project

For the casual user and for developing straightforward applications, the project file is merely the semi-visible repository of options needed to control the making of a program. For the advanced user it can be an entire script to control all and individual compilations, linking and the running of other programs. The project language has a mixture of declarative and imperative statements (including conditional actions) and a macro facility; they are tailored for tasks and objects in the environment. The tasks include compiling, linking, VIDing and running any program. The objects include edit windows and their files, other files and parameters in the environment. System-defined macros yield values set in the environment. The user can define macros too (and recursive ones) to extend the system, for example quickly to carry out non-trivial repetitious actions.

There are two manuals to serve the two types of user: the *Environment User's Guide* gives simple and adequate instruction on how to develop programs and run VID; and the *Developer's Guide* spills the rest of the beans on the project system and its language, command-line operation, pragmas, language mixing and the other advanced features.

The *make* system will sort out exactly what needs to be recompiled. Since the earliest release, I have never seen it caught out by any changes to independent files (including the project file itself). The result of a make - driven by the .PR file - may be .OBJ, .EXE. .LIB, .DLL or even other types produced by a program which you invoke. (Figure 2).

Because the difference in the project system between V2 and V3 is more than an enhancement, old .PRJ files have use only as reminders. I am finding it more fruitful to follow the Developer's Guide and write new .PR files rather than hang on to the old ones. The 'read me' file DOS_ENV.DOC gives a lot of detail on how to convert MAKE files from other systems.

## Linking

The integral linker is worth its own heading because, in combination with other subsystems, there is quite a lot to it. It is 'smart' in that it will only include the data and procedures which are referenced. (You can check this with VID.) The 'smart' approach has been extended to OOP with 'Smart-Method' linking which does the same for classes and methods.

The automatic overlay subsystem is invoked by choosing the Overlay memory model; it is segment based and will swap both code and data. The controls to make user segments resident or discardable are exercised by using a simple overlay configuration file. If finer control is needed at run time then the API can be used to manage overlays and memory and to install error handlers.

## Modula-2 differences

My earlier comments about documentation *(.EXE Magazine, September 1990)* were a bit spiky in parts. The new language tutorial is more approachable and gives an extended essay on low-level features (C programmers watch out! we're coming) but, curiously, leaves out Object Orientation. The *Reference Manual* is a lot better organised and that organisation is more clearly visible.

The OOP features are both taught and illustrated in the *Reference Manual.* I liked the revisions for their clarifications and for the examples; they are more realistic. The OOP features have been extended quite a bit from V2 and now include multiple inheritance, object initialisation and the checked guard operator. This has meant changes to syntax; for example:

- The procedure `Lib.IsOfClass`, used for comparing types at runtime, has been replaced by the operator `IS`.
- The optional qualifier `VIRTUAL` moves from the end to the beginning of a method header.
- The class implementation coding has been extended to make each class more like a Modula-2 module proper.

There is a utility program, NEWCLASS, to convert your source to the new syntax.

The new object initialisation is analogous to module initialisation. I had looked for it in V2, but it wasn't there. It means that it is now not necessary to make a separate call to an initialisation method for each newly created object (ie more like C++ than Turbo Pascal). The initialisation code is executed when an object is instantiated, ie

- Immediately after program loading for a static, module-level object.
- On explicit allocation for a dynamic object referenced by a pointer (use NEW rather than ALLOCATE; the latter will create the object, but won't initialise it).
- On implicit allocation inside a procedure.

The new Modula-2 syntax contains no features for operator/function overloading.

If you were hoping to use Modula-2 for OOP development with some existing class

# New Products ★ Stunning Technology ★ Unbeatable value ★ from REM

REM specialises in designing computers which operate reliably in harsh environments, such as factory floors. To achieve greater reliability REM reduced the number of critical parts in their computers by the use of surface mount technology and a very highly integrated chip set. REM computers typically have 60% fewer points of failure than equivalent computers, this leads to MTBF (mean time between failure) figures approaching 100,000 hours. It is because of this reliability that REM backs its computers with a three year warranty. REM is now offering the high reliability of its computer systems to a wider base of users, with the introduction of its range of microcomputers, for use in office environments.

All four products described below redefine previous price performance levels.

## The REM STATION 1 = £2,495 (486/33 Mhz based workstation)

The original design aim was to provide a usable Windows workstation. The 20 inch screen and the i486 allows the user to effectively multitask between applications, the REM STATION 1 also provides an excellent platform for CAD/CAM, desk top publishing, or as a high performance file server.

*Specifications:*

★ 486/33 Mhz Processor (upgradeable to 50 Mhz)
★ 5.25 + 3.5 inch Floppy Drives
★ High Resolution Graphics (1024 x 768) (Options up to 1600 x 1200)
★ 3 Year Warranty
★ 20 inch colour monitor 1024 x 768 (Options up to 1600 x 1200)
★ 4 MB Ram standard (expansion to 32 MB)
★ 80 MB Hard Disc (expansion to 3GB)
★ Enhanced AT Tactile Keyboard
★ DOS 5. Windows 3 + Microsoft mouse

## The REM STATION 2 = £3,495 (Supercomputer 486/33 + RISC processor)

The REM Station 2 is an ultra high performance workstation for the most demanding applications, such as ray tracing, photo realistic rendering, simulation, and complex CAD/CAM. Super computer like performance is achieved by the use of a unique multi-processing architecture. The REM Station 2 combines the best of CISC and RISC technology. Currently we support the Intel i860 and SPARC chipsets, future planned releases will provide support for the MIPS & MOTOROLA RISC chipsets.

*Specifications*

★ 486/33 Mhz Processor (upgradeable to 50 Mhz)
★ 5.25 + 3.5 inch Floppy Drives
★ High Resolution Graphics (1024 x 768) (Options up to 2024 x 2024)
★ 3 Year Warranty
★ 20 inch colour monitor 1280 x 1024 (Options up to 2024 x 2024)
★ 4 MB Ram standard (expansion to 64 MB)
★ 100 MB Hard Disc (expansion to 3GB)
★ Enhanced AT Tactile Keyboard
★ DOS 5. Windows 3 + Microsoft mouse

## The REM PORTABLE = £1,495 (486/33 Mhz based portable computer)

The REM Portable provides a powerful computer capability for applications requiring mobility.

*Specifications*

★ 486/33 Mhz Processor
★ 3.5 inch Floppy Drive
★ 6.5 pounds gross weight
★ 3 Year Warranty
★ 10 inch paper white LCD screen
★ 2 MB Ram standard (expansion to 16 MB)
★ 40 MB Hard Disc (expansion to 120 MB)
★ DOS 5

## The REM TIGA = £349 (TIGA Graphics accelerator)

The REM TIGA is a graphics accelerator card which provides virtually instantaneous redraw times on even the most complex drawings. It features a TI34010 graphics processor delivering in excess of 6 MIPS.

**If you would like further information or a demonstration please contact the**

# REM Information Hotline on

# 071-436 7438

library, you may be stuck. There is compatibility with objects implemented in the other TopSpeed languages, but it is awkward (you are obliged to make heavy use of pragmas, you have to use C++'s mangled names) and limited (cannot use C++ classes with virtual bases or pointers to members, as there is no Modula-2 equivalent).

The OOP programming features are very complete and elegant. However, JPI supplies no class libraries for Modula-2, and the Rogue Wave library supplied with C++ is not compatible.

## Virtual pointers

The TopSpeed implementation of Modula-2 already has many extensions of its own. Another one, new in this release, is the type `VIRTUAL POINTER`. A virtual pointer (VP) is a part-time pointer - more like a handle - and it may be any type of Modula-2 variable. A VP can have values assigned and compared and it can be passed as a parameter; on these occasions it behaves like a variable. Only when it is dereferenced does it behave like a true pointer.

When a VP is declared, a function procedure is associated with it. (See Figure 3 for the tie-ups.) This function procedure takes one argument, and it is the type of this argument which determines the type of the virtual pointer.

When a VP is dereferenced the function identified in its declaration is called and it returns a `POINTER TO` type. There has to be an actual value for the argument supplied in the call to the function; there is, and it is the VP itself. The procedure can, of course, have side effects as well as deriving the pointer which is its returned value. The address of the VP itself cannot be derived from the actual parameter, only its value(s). (I found all this out with time-consuming experiments and the invaluable VID; the manual only has 10 lines of text on the subject.)

The example supplied, COMPDEMO.MOD, illustrates the workings by implementing a non-fragmenting memory manager; the variables which are the virtual pointers are `CARDINAL` handles to blocks of memory. Their values (in an array) are set by calling other function procedures which return handle values and by using the VP's type to achieve the necessary transfer (old-fashioned 'cast'). Dereferencing a handle means just what it looks like; the associated function converts the handle to a pointer (by doing a table look-up). Notice that the explicitly called `Allocate` and `Deallocate` procedures in the example do the memory management, the implicitly called function just converts handles.

The virtual pointer is like a baby OOP technique which has one static method per class. It uses the same underlying mechanism as object initialisation. VPs could be used to implement virtual storage or associative storage; in a multi-tasked system they could be used to time-stamp or serialise referenced data or to re-display it on the screen if it has changed. If used as a shadow for a regular pointer, a VP would be useful just for its side-effects: to trace or instrument a program.

## Syntax checker

With all languages the short-cut Alt-C starts a compilation. With Modula-2 only, an optional syntax check is initiated. This very quickly detects some errors and saves the time taken by a full-scale compilation if any are present. Only pure syntax is checked, not semantics. If brackets or quote marks are mismatched or punctuation is missing or inappropriate the syntax checker will spot it, but if a name is misspelled or misused it takes a full compilation to find out.

## Windows and PM

The optional TechKit supports development of DLLs (yes *DOS* DLLs) and programs for MS Windows V3 and for OS/2 PM in any of the TopSpeed languages (including Modula-2 - although, being a non-Windows man, I have not tried it). The separate documentation is compact - a combined cook-book and rule-book - which explicitly excludes tutorial material. The toolset comprises a massive .DEF file (equivalent to the C Windows header file) and a resource compiler so that, even if you do need the Windows documentation, strictly speaking you don't need the SDK.

The serious Windows developer would be slowed down without some extra tools. JPI does not yet appear to have committed itself to providing assistance to the Modula-2 programmer in the form of resource editors or application frameworks for icons, Help and GUI objects. C and C++ developers (sniff) are in better shape with the availability of this kind of gear from other suppliers.

## Some cautions

There are signs, little ones, of haste in getting the package together. They show in the documentation and in parts of the context-sensitive language-help. Some of the manuals have Errata sheets and the Modula-2 language 'read me' file DOS_M2.DOC has some of the same corrections to the manual as its predecessor.

When editing a source file, with the cursor on the name of a library procedure, you can key F1 and see the help text which gives the procedure header and description. Some procedures seemed to be missing when I used this method but I found them by going through the Help indexes (for example: `Window.Use` and others in the `Window` module).

I've had a few minor problems:

Some of my programs (but not the TopSpeed demos) hang the system on exit

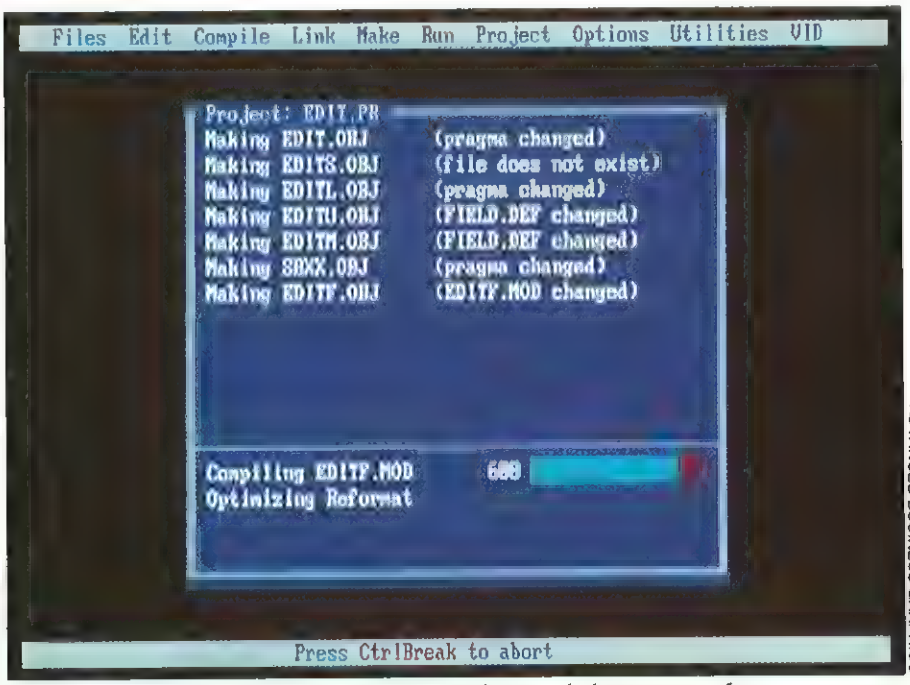
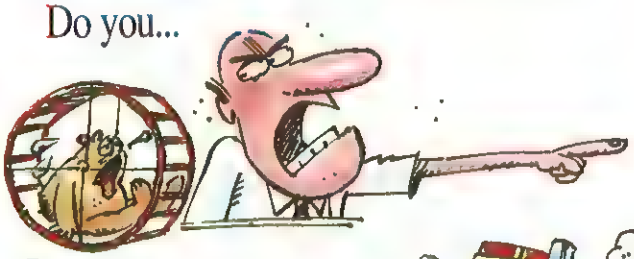


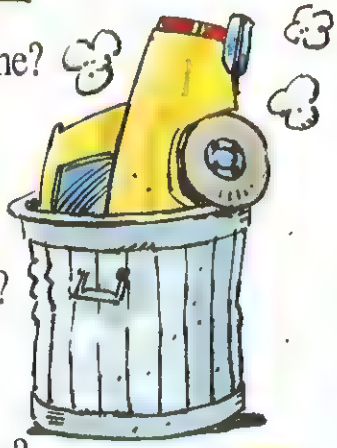

*Figure 2 - You can see why modules are made*

# A SOFTWARE DEVELOPERS Quiz?

**PROBLEM:** Your car needs a tune-up. Do you...

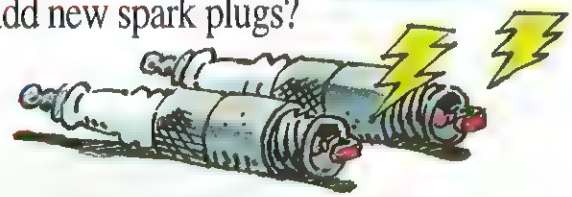


A. Replace the Engine?



B. Replace the car?

C. Add new spark plugs?

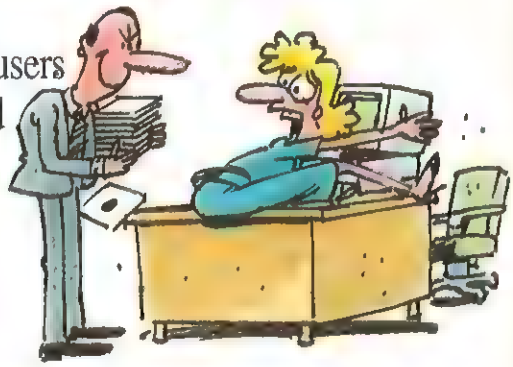


*ANSWER: B is preferable, but C is the logical approach.*
*LOGIC: A tune-up costs less than replacement.*

**PROBLEM:** You've added some changes to your software program. Do you...

A. Make your users get a new computer?

B. Make your users replace the old version with a new version?



C. Let your users simply patch only the changes into the existing version?

*ANSWER: B is the practiced approach, but C is the future of software updating.*
*LOGIC: Patching changes into software systems saves money, time and hassle.*

## INTRODUCING .RTPatch™

.RTPatch lets you send software updates to users with small "patch" files which contain only the differences between the old version and the new version. Works with any type of software file, including EXE, text, data, ...ANYTHING, Windows, DOS, and OS/2®.

MANY POWERFUL OPTIONS • AUTOMATIC VERIFICATION

**45 DAY GUARANTEE**

If you're not completely satisfied with the performance of RTPatch, return it within 45 days from purchase date for a FULL refund of your purchase price.

POCKET SOFT™

**Available *from* Roundhill**
**List Price £295**
**Call 0672-84535**

.RTPatch and Pocket Soft are trademarks of Pocket Soft, Inc. OS/2 is a registered trademark of International Business Machines Corporation.

P.O. Box 821049 • Houston, TX 77282 USA • **Phone:** 713-460-5600 • **Fax:** 713-460-2651

**Roundhill Computer Systems Limited**
Ogbourne St. George, Marlborough, Wiltshire SN8 1SU England
Telephone: 0672-84535, Fax: 0672-84525

AUTHORIZED RESELLER

CIRCLE NO. 140

FULL LOCAL SUPPORT

(with or without HALT) when made with the Small memory model. I used `Lib.Terminate` to circumvent this.

In earlier versions the tiny routine INITEXE, which sets segment address values, was remade if you changed the memory model. In V3 there is only one INITEXE for all models. If you are converting to V3 then make sure that you hide existing INITEXE.OBJ files because they can hang the system. You could change the search rule in the Redirection File ie the one line: `'*.OBJ=.;C:\TS\LIB;'` to: `'*.OBJ=C:\TS\LIB;.;'`) or you could just delete the .OBJs.

```
MODULE TESTVP2 ;
(*  Illustrate tie-ups when *)
(*  using Virtual Pointers. *)

IMPORT IO (* etc *) ;

TYPE
    DataRec =
    RECORD
(*      your data here *)
    END ;
    DataPP      = POINTER TO DataRec ;
    RecHandle   = CARDINAL ;
    KeyedRecPP  = POINTER TO KeyedRec ;
    KeyedRec    =
    RECORD
        RecKey  : RecHandle ;
        Next    : KeyedRecPP ;
        TheData : DataPP ;
    END ;

VAR Head : KeyedRecPP ;  (* linked list an-
chor *)

PROCEDURE HandleDeref (H : RecHandle) : Da-
taPP ;
VAR  This : KeyedRecPP ;
BEGIN
    This := Head ;
    WHILE This <> NIL DO
        WITH This^ DO
            IF  H = RecKey THEN
(*              direct effect *)
                RETURN TheData
            END
        END ;
        This := This^.Next ;
    END ;
(*  side effect *)
    IO.WrStr ("Invalid handle: ") ;
    IO.WrCard (H, 0) ;
    HALT ;
END HandleDeref ;

TYPE
    Key = VIRTUAL POINTER HandleDeref ;

VAR
    MyKey  : Key ;
    MyData : DataRec ;
(*  ... etc for application *)

BEGIN
    Head := NIL ;
(*  application start-up;
    set up handles;
    ...
    ...  ^)
(*  Give virtual pointer a value, eg: *)
    MyKey  := Key (42) ;  (* type transfer
*)
(*  Retrieve the data *)
    MyData := MyKey^ ;
(*  process the record
    ...
    ... *)
END TESTVP2 .
```

*Figure 3 - VIRTUAL POINTERs and their usage*

Occasionally, the first compilation in a session takes a long time to start up; key Ctl-Break and start again. The 'read me' file DOS_ENV.DOC describes how best to use memory to speed the start-up and swap activities of the environment. There may have been some unexpected interactions with my disk cache (but it would be the first time I have had such problems).

## Performance

I rely heavily on the cache in my 2MB Dell 220 and it works well with the TopSpeed environment. First-time start-up takes five seconds with V3 instead of four with V2 and the time for swapping - for example to do a VID run - is about the same. Using equivalent compile and link options, V3 .EXE files are typically smaller than V2 and times for make are less; just a few percent in each case. Run times are the same in all the cases I could compare, except for my number-crunching program which was a few percent slower.

## Multi-language

TopSpeed C language is ANSI X3 conformant and BSI validated; C++ is implemented to the standard set by AT&T

# THE SOFTWARE CONSTRUCTION COMPANY

## BORLAND

| | |
|---|---|
| Borland C++ | £199 |
| Turbo C++ 2nd. Edition | £55 |
| Turbo Debugger & Tools | £85 |
| Turbo Pascal v.6.0 | £69 |
| Pascal Professional | £129 |
| Pascal for Windows | £99 |
| ObjectVision | £79 |
| Paradox 3.5 | £335 |
| Paradox Engine | £199 |
| Paradox SQL Link | £215 |

## AUTHORISED

### DATABASES

| | |
|---|---|
| Advanced Revelation | £795 |
| Advanced Revelation - | £695 |
| askSam | £258 |
| askSam LAN | £579 |
| Clarion Personal Dev. 2.0 | £79 |
| Clarion Professional Dev. | £679 |
| DataEase | £485 |
| DataEase - LAN 3-PAK | £485 |
| Dataflex | £449 |
| Dataflex Multiuser | £1065 |
| Knowledgeman/2 | £645 |
| Magic PC | £325 |
| Open Access III | £449 |
| Open Access III Compiler | £385 |
| R:BASE 3.1 | £515 |
| R:BASE Compiler | £190 |
| R:BASE 3.0 LAN Pack | £645 |

### DBASE

| | |
|---|---|
| Clipper 5.0 | £325 |
| dBASE III PLUS | £395 |
| dBASE IV | £495 |
| dBASE IV Developer's Ed. | £695 |
| dBFast/DOS | £255 |
| dBFast/PLUS | £255 |
| dBXL | £165 |
| dBXL LAN | £389 |
| Force | £449 |
| FoxBASE+ | £295 |
| FoxBASE+/386 | £425 |
| FoxPro | £515 |
| Vulcan | £320 |

### OBJECT-ORIENTATED

| | |
|---|---|
| Smalltalk/V | £65 |
| Smalltalk/V 286 | £129 |
| Smalltalk/V Windows | £325 |

### CASE TOOLS

| | |
|---|---|
| EasyCASE Plus | £190 |
| EasyCASE Plus Prof. | £255 |
| EasyCASE Prof. Upgrade | £80 |
| Visible Analyst | £385 |
| Visible Analyst Workbench | £2460 |
| Visible Analyst Workbench | £6995 |
| Visible Analyst Workbench | £1295 |
| Visible Dictionary | £595 |
| Visible Prototyper | £595 |
| Visible Rules | £595 |

### VERSION CONTROL

| | |
|---|---|
| MKS RCS 5.1 | £165 |
| MKS RCS 5.1 - 5 User | £645 |
| PVCS Professional | £319 |
| PVCS - 5 User | £1345 |
| SMS | £395 |
| Sourcerer's Apprentice | £175 |
| Sourcerer's Apprentice | £429 |

### TEXT AND UNIX TOOLS

| | |
|---|---|
| MKS AWK | £65 |
| MKS AWK DOS & OS/2 | £129 |
| MKS Programmer's Platform | £429 |
| MKS Toolkit | £165 |
| PolyAWK | £129 |
| PolyAWK Toolkit | £145 |
| SpellCode | £65 |

### TRANSLATORS

| | |
|---|---|
| BASTOC | £515 |
| BAS_C Commercial | £579 |
| dBx/dBPort | £389 |
| dBx/dBPort with source | £645 |
| FOR_C | £359 |
| Pascal to C Translator | £99 |

## Microsoft

| | |
|---|---|
| BASIC PDS 7.1 | £219 |
| Visual BASIC | £99 |
| C 6.0 Compiler | £210 |
| C 6.0 & Windows SDK | £349 |
| COBOL 4.0 | £345 |
| FORTRAN 5.1 | £199 |
| MASM 6.0 | £65 |
| MS-DOS 5 Upgrade | £59 |
| Pascal Compiler | £149 |
| QuickBASIC | £50 |
| QuickC | £50 |
| QuickC/Assembler | £85 |
| QuickPascal | £50 |
| Source Profiler | £49 |
| Windows 3.0 | £65 |
| Windows S.D.K. | £260 |

*Authorised*

### EDITORS

| | |
|---|---|
| BRIEF for DOS & OS/2 | £199 |
| BRIEF upgrade | £60 |
| BRIEF with dBRIEF | £275 |
| BTags | £35 |
| Cheetah | £159 |
| dBRIEF for dBASE/Clipper | £99 |
| dBRIEF for Paradox | £99 |
| dBRIEF for R:BASE | £99 |
| EDT+ | £190 |
| EMACS | £210 |
| Epsilon | £129 |
| KEDIT | £99 |
| MKS Vi | £99 |
| Multi Edit Professional | £115 |
| Norton Editor | £50 |
| SPF/PC | £159 |
| SYNDIE | £319 |
| VEDIT PLUS | £120 |

### LINKERS

| | |
|---|---|
| .RTLink | £190 |
| .RTLink/Plus | £319 |
| ALINK | £129 |
| Blinker | £175 |
| Plink w/LTO | £359 |

### MAKE UTILITIES

| | |
|---|---|
| MKS Make | £99 |
| MKS Make - 5 User | £320 |
| PolyMake | £129 |
| PolyMake - 5 User | £485 |

### PARSERS AND LINTS

| | |
|---|---|
| MKS Lex & Yacc | £165 |
| PC-Lint | £90 |

### PROFILERS

| | |
|---|---|
| .RTLink/Plus | £319 |
| CHARGE | £65 |
| Codesifter | £79 |
| Inside! | £85 |
| PC Metric | £129 |

### DEMO BUILDERS

| | |
|---|---|
| Dan Bricklin's Demo II | £190 |
| Instant Replay III | £99 |
| Instant Replay Prof. | £385 |
| Show Partner F/X | £259 |

### MEMORY MANAGERS

| | |
|---|---|
| 386 MAX Version 5 | £85 |
| Above Disk | £79 |
| Blue Max | £100 |
| HEADROOM | £85 |
| HEADROOM Network Extens | £39 |
| HI386 Complete | £65 |
| MOVE'EM | £59 |
| QEMM-386 | £65 |
| QEMM-50/60 | £65 |
| QRAM | £55 |
| Turbo EMS | £85 |

### DOCUMENTATORS

| | |
|---|---|
| 4c for Brief | £79 |
| ABC Flowcharter | £190 |
| ASMFLOW | £129 |
| C-Clearly | £85 |
| C-Doc | £120 |
| Clear for C | £129 |
| Clear for dBASE | £129 |
| dAnalyst for C and C++ | £190 |
| dAnalyst for Clipper | £190 |
| dAnalyst for dBASE III | £190 |
| dAnalyst for dBASE IV | £190 |
| dAnalyst for FoxBASE + | £190 |

## FREE

## Microsoft Flight Simulator

*WORTH £45 WITH EVERY PRE-PAID ORDER OVER £100**

| | |
|---|---|
| EasyFlow | £99 |
| Flow Charting III | £165 |
| SourceDoc | £190 |
| The Documentor | £190 |
| Tree Diagrammer | £65 |

### DISASSEMBLERS

| | |
|---|---|
| Sourcer 486 | £85 |
| Sourcer w/BIOS pre-proc. | £110 |

### E-MAIL

| | |
|---|---|
| cc:Mail | £450 |
| cc:Mail EXPAND | £385 |
| cc:Mail Remote | £190 |
| DaVinci eMAIL | £645 |
| DaVinci eMAIL Combo | £965 |

### CODE GENERATORS

| | |
|---|---|
| DATABOSS | £395 |
| Matrix Layout | £195 |
| ObjectVision | £259 |
| ObjectVision for Windows | £259 |
| PRO-C | £259 |
| R&R Code Generator | £99 |
| UI2 Developer's Release | £385 |
| UI2 Touch & Go | £259 |
| Zachary | £259 |

### DEBUGGERS

| | |
|---|---|
| Periscope 286/386 Pod | £389 |
| Periscope I OK | £319 |
| Periscope I/512K | £385 |
| Periscope I/MC | £449 |
| Periscope I/MC 512K | £449 |
| Periscope II | £145 |
| Periscope II-X | £129 |
| Periscope IIw/Switch | £145 |
| Periscope IV (16 MHz) | £775 |
| Periscope IV (25 MHz) | £900 |
| Sherlock | £129 |
| SVS DBG | £210 |

### MULTI-TASKING

| | |
|---|---|
| DESQview 2.2 | £85 |
| DESQview 386 | £125 |
| Dr. Switch-OnCall | £65 |

### BATCH ENHANCERS

| | |
|---|---|
| Batcom | £39 |
| Beyond.BAT | £65 |
| Command Plus | £85 |
| ProKey Plus | £85 |
| The Builder | £99 |

### TERMINAL EMULATORS

| | |
|---|---|
| BLAST II | £195 |
| COTERM/100 | £145 |
| COTERM/220 | £160 |
| COTERM/220 | £160 |
| COTERM/4010 | £165 |
| EM 320 | £199 |
| EM 4010 | £230 |
| EM 4105 | £330 |
| EM 4105 Network | £345 |
| ICE.10 | £190 |
| SmarTerm 125 | £190 |
| SmarTerm 2392 | £65 |
| SmarTerm 240 | £225 |
| SmarTerm 320 | £129 |
| SmarTerm 340 | £225 |
| SmarTerm 400 | £115 |
| SmarTerm 4014 | £145 |
| Smarterm 470 | £190 |
| ZSTEM 100 | £65 |
| ZSTEM 220 | £99 |

### PROFESSIONAL SHAREWARE FOR DEVELOPERS

| | |
|---|---|
| ASM Library (20) | £40 |
| C/C++ Library (68) | £66 |
| dBASE/Clipper (87) | £100 |
| DOS Utilities (37) | £40 |
| Novell Netware (54) | £66 |
| Turbo Pascal (38) | £40 |
| Windows (64) | £66 |

**(Number of diskettes)**

*All prices exclude V.A.T. U.K. carriage included. ***One offer at time of purchase, per order at published prices.*** VISA or ACCESS accepted. Credit to approved accounts.

**SUMMER '91 CATALOGUE**

Call for your free copy of our 160 page catalogue, TOOLBOX.

**1 THE MALTINGS, GREEN DRIFT, ROYSTON, HERTS. SG8 5DB**
**TELEPHONE (0763) 244114 FACSIMILE (0763) 244025**

version 2.1. Both of these have extensions including based pointers. Pascal is at ISO level 1 but has extras - to make it more modern - including, of course, separate compilation of units.

In this release JPI has given a lot of attention to what the rest of the world is doing. There is extensive documentation on how to use other vendors' libraries and - as real examples of going outside - there are a Modula-2 .DEF file for Borland Graphics, a Rogue Wave class library for C++ and a utility to prepare programs for debugging using CodeView. JPI has introduced the 'TopSpeed Consortium' through which invited suppliers are already offering other add-ons in the form of libraries.

Within the JPI collection, compatibility is ensured. With other libraries, care has to be taken to choose the right memory model (generally Large) and calling convention (on the stack rather than in registers); these choices are exercised using pragmas. Multi-language OOP applications are possible too, but need a touch of extra care since the C++ and Modula-2/Pascal objects are not entirely compatible. The Developer's Guide gives details of the rules and it publishes the memory layouts for objects.

The environment is not restricted to the four languages offered by JPI (plus assembler if you take the TechKit). With the project system and with menu tailoring the environment is open-ended to the extent that all kinds of utilities - including new compilers - can be plugged in.

***Be aware of the cost of runtime checks***

## Solid

The key technical features of this release are the new, mixable programming languages and introduction of the more powerful project system as part of the environment. The proven TopSpeed code generator and the environment make it worth taking a serious look. For me, the main improvements to this product are in the documentation, which is better written and better organised, and in the project system which, now more exposed to view, has a more understandable role.

The key commercial features are the same as the technical ones plus a mix and match approach to buying the bits; you choose just which languages and tool kits you want. JPI has taken a bold step in competing in the C, C++ and Pascal compiler market, particularly as there is substantial support for Windows development elsewhere. For straight Modula-2, there is little competition.

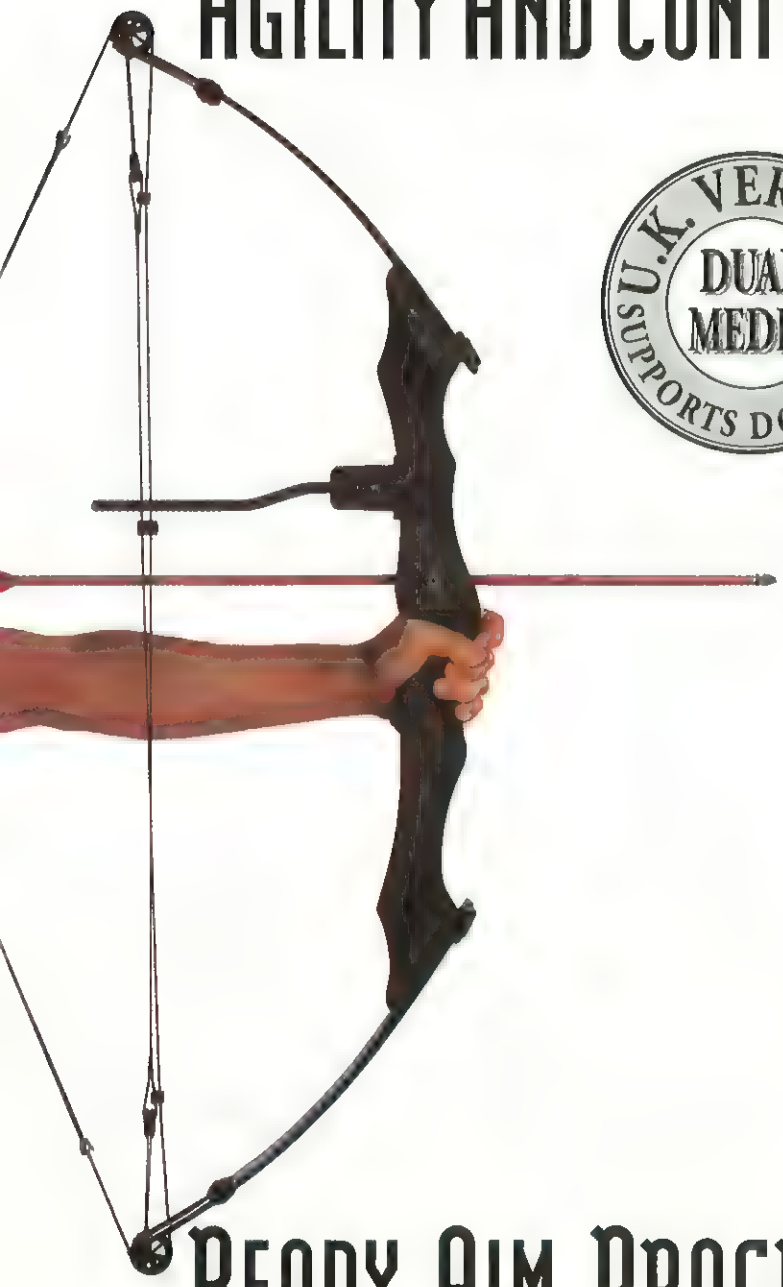
EXE

---

*Richard H Pickard has been a full time data processing practitioner for many years and is equally interested in the application of DP techniques to business problems as in the techniques themselves. He has used all the releases of TopSpeed Modula-2 since 1.04. Richard may be contacted on 0525-61836.*

*The Environment, the compilers, the TechKit and library source kits are all priced £59 each (plus VAT). Jensen & Partners (UK) Ltd are at: 3 The Mansards, Tavistock Street, Bedford, MK40 2RX; 'phone 0234 267500.*

# BRIEF'S NEW MOUSE SUPPORT GIVES YOU GREATER AGILITY AND CONTROL.



U.K. VERSION
DUAL MEDIA
SUPPORTS DOS & OS/2

More than 100,000 programmers around the world swear by the virtues of BRIEF – its unmatched power, flexibility, and intuitive command structure. Now, thanks to mouse support and other new features, version 3.1 of the best-selling text editor will make you quicker and more productive than ever. Hit a button on the mouse – a special, user-configurable menu window pops up and lets you execute any command you wish. Mark text and move around in your windows without using the keyboard. Zoom your windows to fill the screen. As snappy as it is, the mouse is only one enhancement among many. BRIEF 3.1 also takes advantage of EMS memory whenever it reads in a file. The result: faster performance, whether you're editing a file or running a macro. So what else is new? Redo – a step for step complement to undo. Support for Microsoft C 6.0 Advisor and Microsoft's Programmer's Workbench. And more. BRIEF's seamless interface with other Solution Systems products gives you instant access to the tools you need. A single keystroke lets you keep track of product development with Sourcerer's Apprentice – Version Control for the Professional. Analyze product performance with CHARGE. And nothing beats dBRIEF for tailoring BRIEF to Dbase or Paradox.


*BRIEF sells for £199 excluding VAT.*

*Once you start working with BRIEF 3.1 you'll really fly.*

# READY, AIM, PROGRAM.

*Order your copy of BRIEF today.*

*call:*

**0763 244141**



**Solutionsystems**

THE PHYSICS OF PROGRAMMING

1 The Maltings, Green Drift,
Royston, Hertfordshire SG8 5DB
Telephone: 0763-244141
Facsimile: 0763-244025

CIRCLE NO. 145

# Personal Supercomputing

*When you need more raw power than a modest PC can provide, you'd normally upgrade to a 386/486. Cliff Saran has been looking at an alternative.*

Have you ever written an application that really pushes your PC to its limits? There are enumerable tasks that are so computationally intensive that running them on a PC should only be attempted if you don't wish to use your machine for the next couple of hours, or it's Friday afternoon, or whatever. Although the 80x86 microprocessor family contains some fast processors, even a 386/486 based PC will groan under the strain. What is needed is a dedicated processor that can perform all the really difficult stuff, leaving the PC free to handle I/O. There are two options open to the aspiring developer; either use a super-fast RISC processor like the 88000, or alternatively, use an array of slower processors. INMOS has adopted the latter approach and its transputer family provides an ideal hardware medium for implementing parallel algorithms although, until recently, software development was limited to occam and so applications only appeared in the supercomputing arena. INMOS now ships a parallel C tool kit with a C compiler that is ANSI certified; this should allow developers using conventional workstations to port their existing, sequential applications to the transputer and so reap the benefits of parallel processing.

## The B008 mother-board

You may recall that the transputer is simply a microprocessor with four high speed serial connections (Link0 to Link3) to the outside world. These bidirectional **Links** enable data to be exchange between transputers in a network at speeds of up to 20 Mbits per second. The IMS B008 **TRA**nsputer **M**odule (TRAM) mother-board from INMOS is a typical transputer card that can be plugged into a PC or AT bus. It can accommodate up to 10 TRAMs. Each TRAM is a small printed circuit board that plugs into one of the slots (numbered one to ten) in the mother-board, and consists of a single transputer and either one or two megabytes of on-board memory. INMOS supplies TRAMs for its entire transputer family and it is possible to mix and match different transputers on the same mother-board. Our evaluation board was supplied with two **T800** TRAMs and provisions were made for communication with a second TRAM mother-board using an edge connector. An array of transputers can be connected either directly, in the hardware, or using the **IMS C004** link switch that enables Link0 and Link3 of any transputer to be connected to any other under software (see Figure 1). Once such a network has been *softwired*, it will behave just as if it had been *hardwired*.

## Installing the B008

After a quick browse through the hardware set up manual, my understanding of the mother-board was far less than it was before I had opened the boxes. Fortunately there was an example given, illustrating the board settings required to set up a single TRAM. On the mother-board, each TRAM is connected to its neighbour through a pipeline via links one and two (see Figure 1) and for this pipeline to be maintained, links one and two of any unoccupied slot on the mother-board must be connected using a *pipejumper*. Slot one is special and is reserved for the **Root** transputer. This must always be present since it is the only transputer on the mother-board that can talk directly to the PC bus. The Root TRAM contains three additional pins which must be inserted on the under-side of the module to allow the network of transputers to access initialisation and debugging *System Services*. Additional TRAMs can now be inserted in any of the vacant slots (although logically TRAM2 should go into Slot2 and so on), and the pipeline should be maintained by removing or inserting pipejumpers as appropriate. The **S708DRIV.SYS** device driver must then be installed in order for the PC to talk to the B008 mother-board.

```
DEVICE=S708DRIV.SYS /N LINK1
  /A 150 /D 3 /I 3
```

This puts the B008 at address 0x150 and the other parameters inform it to use DMA channel 3 and interrupt level 3. The tool kit must be told to use the device named LINK1 and this is achieved by setting the `TRANSPUTER` environment variable.

```
SET TRANSPUTER=LINK1
```

## Parallel Processing in C

Without a shadow of doubt, the most powerful feature of the transputer architecture is its ability to distribute an application over a

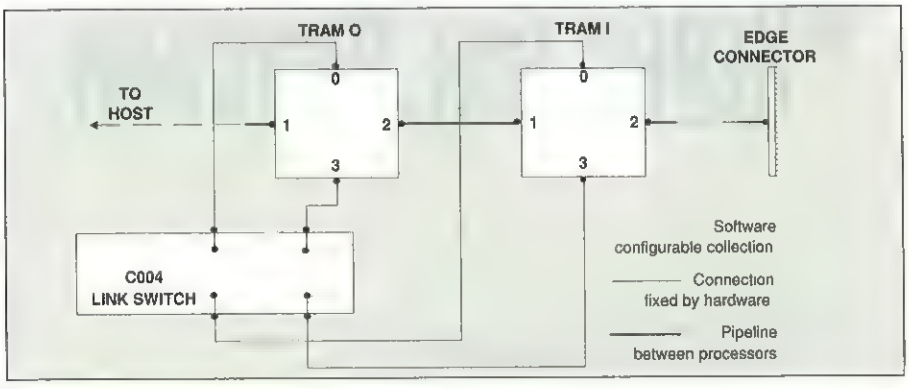



*Figure 1 - Simplified diagram of B008*

```
/* Two T800 connected in a pipeline */
/* Hardware description.*/
T800 (memory = 1M) root;
T800 (memory = 1M) tram1;

/* Host link connection. */
connection RootToHost, RootToTram1;

connect
  root.link[0] to host
  by RootToHost;

connect
  root.link[2] to tram1.link[1]
  by RootToTram1;

*  Software description */
/* 2 Mand() Processes in system. */
val ProcNum 2;

* I/O edges - interface to ISERVER. */
input HostInput;
output HostOutput;

* Process descriptions. */
process (stacksize=20K, heapsize=50K,
  interface (input HostInput,
             output HostOutput,
                    input ToMux[ProcNum],
             int Count=ProcNum,
             output FromMux[ProcNum]
            )) Mux;

process (stacksize=20K, heapsize=50K,
  interface (output ToMux,
             input FromMux
            )) Mand[ProcNum];
```

```
/* Host channel connections. */
connection FromHostInput;
connection ToHostOutput;
connection MuxFromMand[ProcNum];
connection MuxToMand[ProcNum];

onnect
  Mux.HostInput to HostInput
  by FromHostInput;

onnect
  Mux.HostOutput to HostOutput
  by ToHostOutput;

* Mux channel connections. */
rep i = 0 to ProcNum - 1
{
  connect
    Mux.ToMux[i] to Mand[i].ToMux
    by MuxFromMand[i];

 connect
    Mux.FromMux[i] to Mand[i].FromMux
    by MuxToMand[i];
}

/*  Mapping description */
use "main.lku" for Mux;
rep i = 0 to ProcNum - 1
{
 use "Mandproc.lku" for Mand[i];
}

Place Mux on root;
place Mand[0] on tram1;
place Mand[1] on root;
place ToHostOutput on RootToHost;
Place FromHostInput on RootToHost;
```

*Figure 2 - .CFS Configuration file for two processors*

network of processors. This is the macro level of parallelism. Internally, each transputer contains a highly efficient scheduler, implemented in the hardware, that enables a number of independent processes to be run on the same CPU. At both the macro and micro level, inter-process communication is achieved through the use of I/O channels which map onto real transputer links. It is in fact possible to develop an application on a single processor which can later be upgraded to a transputer network without the need to recompile any of the source code. A process that is waiting on a resource such as a timer event (or a channel) does not consume any of that valuable CPU time. Channel I/O occurs concurrently with all other processing activity and data can be transferred over hardware links without the intervention of the CPU.

Using the above hardware model, the INMOS ANSI C compiler conforms to the *Communicating Sequential Processes* (CSP) theory of parallelism based on the idea of independently executing processes that exchange data via unidirectional *channels*. A number of object modules can be linked together producing a Linked Unit (.LKU) which is equivalent to a C main program. A Linked Unit can contain only a single transputer process, although an application may have any number of processes, so each of these will require a separate C main program. Linked Units must be combined to form a single transputer executable file that can then be downloaded onto the transputer network. This is a two-stage process consisting of *configuration* (for informing the network where each process will run) followed by *collection*, where all the relevant files are combined together. Configuration and collection are achieved using the ICCONF and ICOLLECT tools respectively. The result of all this processing is a transputer executable file which is given the extension **.BTL** (Bootable Transputer Link). This file may now be downloaded onto then network using a special loader program (ISERVER.EXE) that runs on the PC. The ISERVER uses the configuration information to program the C004 Link Switch (ie to set up the logical Link connections between transputers) and then places each Linked Unit on the network.

The C run-time library provides a number of functions to manipulate **Process** and **Channel** data structures. The first obstacle to overcome when using concurrent C is that there is a three level hierarchy of process and channel control. This layering cannot be easily grasped from the examples given in the tool set manual. At the highest level of the hierarchy, there are functions to manage instances of processes and channels.

```
/* Create a Process. */
Process* ProcAlloc()
/* Create a Channel. */
Channel* ChanAlloc()
```

Child processes spawned using `ProcAlloc()` can only communicate with the parent or other child processes in the same Linked Unit using Channels created by `ChanAlloc()`; they will only work within this single Linked Unit. This means that child processes cannot be created dynamically over a transputer network. The CSP paradigm requires that processes should be independent and a process created dynamically using `ProcAlloc()` cannot be placed on a second transputer, since the child process would share the same physical address space as its parent. TRAM modules contain no shared memory and so the only method of communication that they offer is through channels. If an application contains only one Linked Unit (ie one parent process), it is said to be unconfigured and will only run on a single transputer.

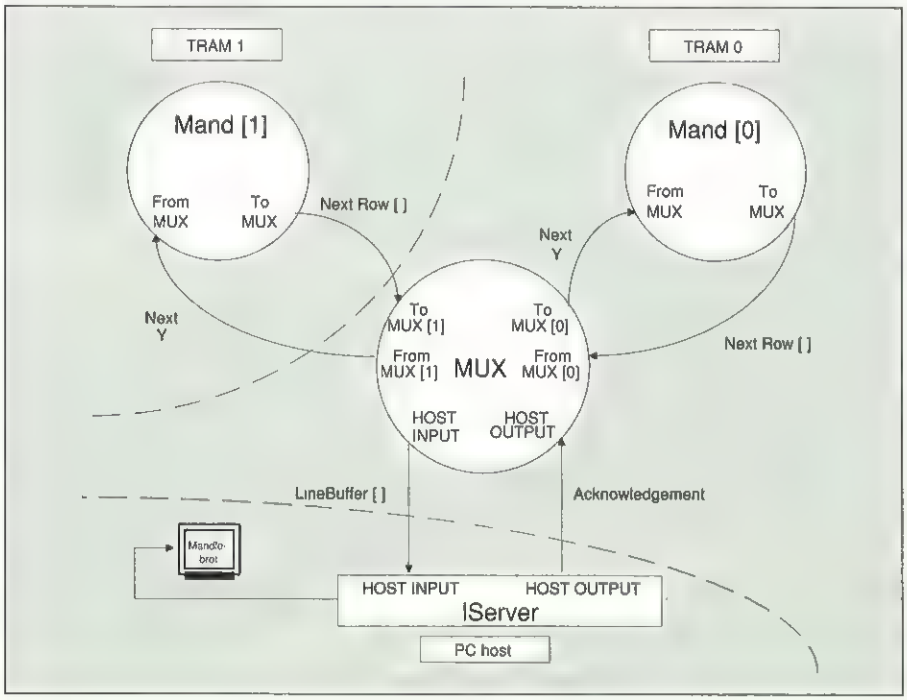



*Figure 3 - Mandlebrot Channel communications*

ICCONF will automatically produce the relevant information for the loader.

So how is a connection made between two processes running on separate transputers? This is where the second level of the hierarchy comes into play. In order to write an application with distributed processes, it is necessary for the network of transputers to be configured manually. ICCONF can take configuration details from a text file (.CFS) and produces *Softwire* information, which is used by the ISERVER loader to initialise the IMS C004 link switch. The CFS file contains two sections. The hardware configuration describes how each processor is connected in the network (a pipeline in the two processor example, with the Root connected to the PC host); the software configuration specifies all the processes running on the network and all their interprocess connections (Channels). As each parent process is a separately compiled and linked C main program, distributing processes over a network is achieved by **placing** a process on a given processor and informing the configurer of which Linked Unit to **use** for that process (see Figure 2).

At run-time it is possible for a process to extract information from the configuration file using the function `get_param(n)`. This is essential if parent processes need to transfer data to one another. As this can only be achieved using Channels, a process can use `get_param` to return a pointer to the Channel specified in the CFS file. Constants may also be extracted (eg in Figure 5, the integer `NumOfProcs` in `main()` is assigned the value of two).

You can't transmit data down one channel without having a corresponding receiver channel on the other end; each output Channel must send a specified number of bytes to an input Channel on another process. It is an error to send or receive more bytes than the other Channel of the pair is expecting and, if this occurs, the offending process will crash. This seems a little overdelicate. Perhaps INMOS could provide an additional layer of software that would allow a programmer to harness the power of Channel communication without having to worry about whether the transfer of data was successful. A short-term solution would be the use of some kind of checking tool that could verify the number of bytes sent or received in each transfer.

Finally, the lowest level through which a process can communicate is achieved by talking directly to the ISERVER. This is only available when using a configured program. Up until now, the ISERVER has not been given much coverage however, without the ISERVER, no application developed using this tool set could ever be executed successfully.

The ISERVER provides two fundamental services for all transputer programs; it first configures the network, as dictated by the CFS file and downloads the program via LINK0 on the Root transputer; it then provides all run-time support for the Host's (in this case the PC's) I/O resources. For instance, when a `printf()` is issued by the transputer code, a write to `stdout` is executed by the ISERVER. Figure 3 illustrates the arrangement of processes in a Mandlebrot drawing application, consisting of three processes where the **Mux** process (running on TRAM0 - the Root) multiplexes input from the two Mandlebrot processes (each running on a different processor). Each `Mand[]` process works on a single row of pixels and then transmits the resulting row to the `Mux` process. This row consists of four graphics planes that are then sent separately to the ISERVER which writes the row into the PC's framebuffer. The code for the multiplexing process is given in Figure 5.

## Talking to the ISERVER

In principle, the ISERVER uses a relatively straightforward protocol for talking to a process on the network. INMOS includes the C source code for the ISERVER with the tool kit, which I was able to recompile with Microsoft C V6.0.

There are two main reasons for looking at the source code. The first is that the run-time library does not support all the functions needed for communication with a PC. However the ISERVER contains a large number of undocumented sub-functions which may be accessed indirectly. The second is just plain curiosity. If such a sub-function is needed, it is essential to understand exactly what the ISERVER is expecting to receive and what it will transmit back to the Root processor.

The protocol is based on a variable length message buffer where the first two bytes contain the length of the message buffer. The first byte of the message is a `Tag` which is used by the ISERVER to determine the type of system service that is required. A `switch` statement in the main loop of the ISERVER (iserver.c) uses this `Tag` to select a built-in function. Sub-functions (eg `spMsdos()` in msdosc.c) are specified using the next byte in the message and parameters that are required by these sub-functions are next, followed by the actual data. Figure 4 shows the message format expected by the `DosPortWrite()` sub-function where a two byte `Length` field is followed by a five byte message. In Figure 5 the function `outp` shows how to invoke the ISERVER's `DosPortWrite()`. Additional commands can be added to the ISERVER by simply creating new tags and writing the appropriate code. In fact, along one of the corridors at INMOS, there's a clever dude who is actually writing an ISERVER interface to the Microsoft C graphics library.

## Debugging

Under MS-DOS, when a program behaves unexpectedly, you normally resort to a symbolic debugger. This is all very well on a primitive processor like the 8086, but if your application consists of many distributed processes then debugging is far tougher. This is why the IDEBUG tool, supplied with the tool kit, has two modes of operation. When using it for *post mortem* debugging, an application is executed until an error (or a CTRL-BREAK) occurs, then the IDUMP utility is used to obtain a core dump from all processors. The symbolic information within the core file can then be examined using IDEBUG.

In the interactive mode, the debugger runs on the root transputer and analyses the state of the network at run-time. This implies that an extra transputer is required to use the debugger so the CFS file must be altered to get it going. Once these changes have been made, IDEBUG provides a sophisticated run-time debugging environment. Breakpoints can be set in any process and variables or pointers can be examined using the C naming convention. Channel and Process pointers have special significance to the debugger and so, when dereferencing a Channel pointer, it indicates the state of the Channel and whether it is empty. If the running process invokes a `ChanOut()` then the debugger ingeniously switches context into the receiving process where the transmitted data can then be examined. It is even possible to *BACKTRACK* from a `ChanIn()` to where the corresponding `ChanOut()` occurred.

The User Interface is not as pretty as CodeView and a windowing environment would be most welcomed. However, once the main facilities have been learnt, IDEBUG provides an intuitive method of debugging con-

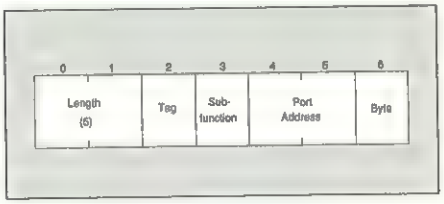
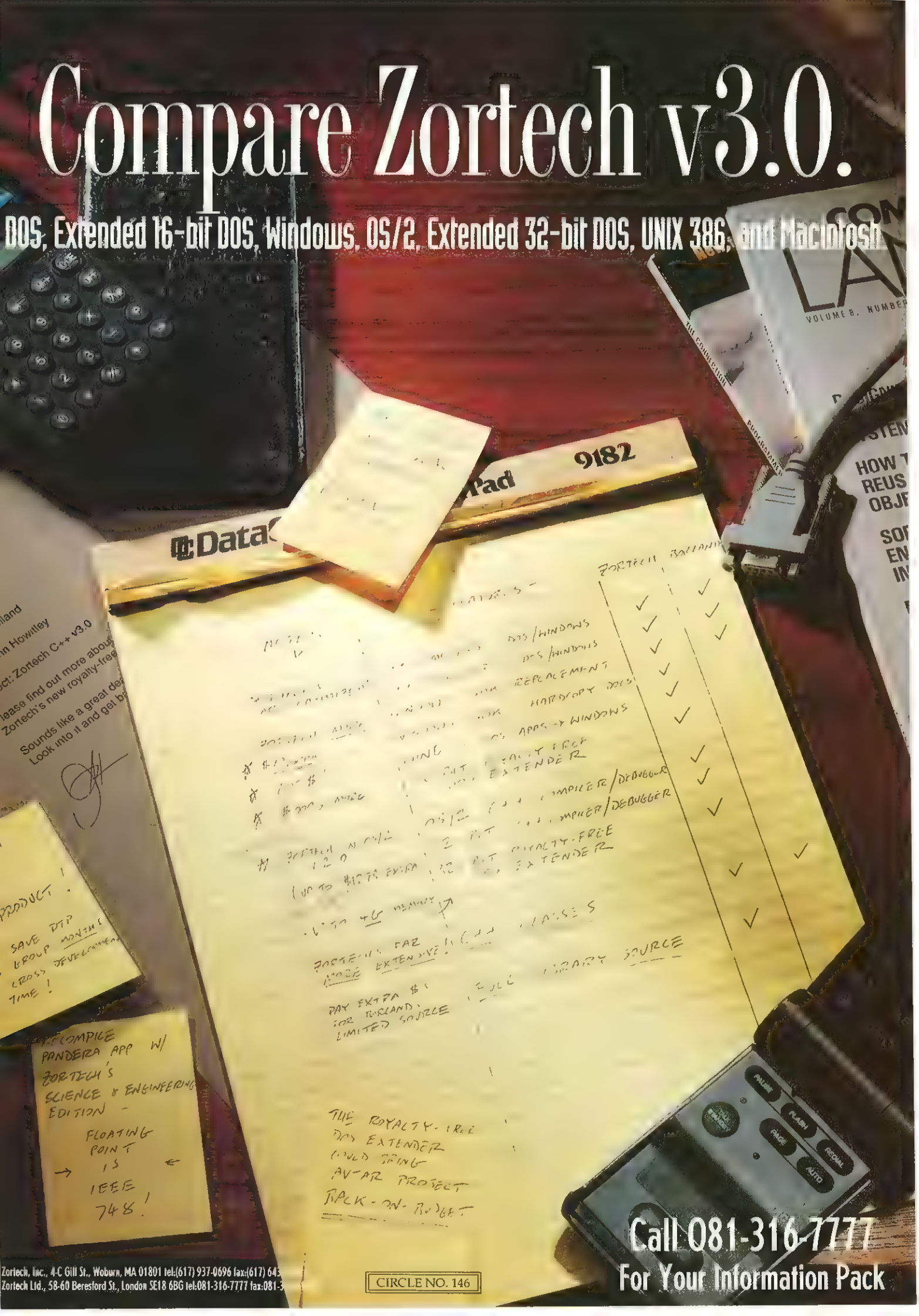



*Figure 4 - ISERVER protocol for DosPortWrite()*

# Compare Zortech v3.0.

## DOS, Extended 16-bit DOS, Windows, OS/2, Extended 32-bit DOS, UNIX 386, and Macintosh




Zortech, Inc., 4-C Gill St., Woburn, MA 01801 tel:(617) 937-0696 fax:(617) 643
Zortech Ltd., 58-60 Beresford St., London SE18 6BG tel:081-316-7777 fax:081-3

```
/* MAIN.C */
/* Main Mux Process. */
#include "main.h"

/* EGA Page0/Mode 16 frame buffer.*/
int pScreen = 0xA0000000;

/* PC I/O port interface. */
void outp ( int Port, char Value)
{
  char pTxBuffer[8];
  char pRxBuffer[5];
  Channel *pOutStream;
  Channel *pInStream;

  /* Get I/O edges.*/
  pInStream =
    (Channel *)get_param(HOST_INPUT);
  pOutStream =
    (Channel*)get_param(HOST_OUTPUT);

  /* Buffer needed by DosPortWrite.*/
  pTxBuffer[0] = 6;
  pTxBuffer[1] = 0;
  pTxBuffer[2] = SP_MSDOS;
  pTxBuffer[3] = DOS_PORT_WRITE;
  pTxBuffer[4] = (char)(Port % 256);
  pTxBuffer[5] = (char)(Port / 256);
  pTxBuffer[6] = Value;
  pTxBuffer[7] = 0;

  /* Transmit buffer.*/
  ChanOut(pOutStream,
          (void *)&pTxBuffer[0], 8);

  /* Receive status information.*/
  ChanIn(pInStream,
         (void *)&pRxBuffer[0], 3);
}


/* Send buffer to ISERVER.*/
void GetRow (int BitPlane, int Row,
             int Column, int Length,
             char pLineBuffer[])
{
  /* Access map mask register.*/
  outp(0x3C4, MAP_MASK_REG);

  /* Enable bit plane.*/
  outp(0x3C5, (char)BitPlane);

  /* Send buffer to screen memory.*/
  to86(Length, pLineBuffer,
       (pcpointer)(pScreen + Column +
       (Row * NUM_OF_COLS)) );
}

/* Virtual framebuffer[Plane][Column].*/
static unsigned char
  LineBuffer[BIT_PLANES][NUM_OF_COLS];
```

```
/* Display a single row of graphics.*/
void SendRow (int Row)
{
  int i;

  for (i = 0; i < 4; i++)
  {
    GetRow((0x01 << i), Row, 0L,
           NUM_OF_COLS,
           (char *)&LineBuffer[i][0]);
  }
}

/* Change Screen Mode.*/
void SetMode (char Mode)
{
                    union REGS reg;

                    reg.h.ah = 0x00;
                    reg.h.al = Mode;
                    int86(0x10, &reg, &reg);
}

/* Change Display Page.*/
void SetPage (char Page)
{
                    union REGS reg;

                    reg.h.ah = 0x05;
                    reg.h.al = Page;
                    int86(0x10, &reg, &reg);
}

/* Create a list of channel pointers.*/
static Channel **
  CreateChannels (
                  Channel *Old[],
                  int Size
                  )
{
  Channel **New = NULL;

  if ((New =
      malloc((Size+1)*sizeof(Channel*)))
      == NULL)
    abort();
  else
  {
    int Count = 0;

    while (Count++ < Size)
    New[Count-1] = Old[Count-1];
    New[Size] = NULL;
  }
  return(New);
}

/* Multiplexor process.*/
int main (void)
{
```

```
int y=0;
Channel **pToMux;
Channel **pFromMux;
int     NextRow;
int     Index;
int     NumOfProcs;

/*  Fetch parameters from CFS file.*/
NumOfProcs =
*(int *)get_param(NUM_OF_PROCS);

pToMux = CreateChannels(
get_param(MUX_FROM_MAND), NumOfProcs);

pFromMux = CreateChannels(
get_param(MUX_TO_MAND), NumOfProcs);

SetMode(0x10);
SetPage(0);
while (1)
{
  /* Wait for signal from Mand().*/
  Index = ProcAltList(pToMux);
  NextRow =  ChanInInt(pToMux[Index]);

  /* Check if Mand() is requesting *.
  /*  a new row to process.*/
  if (NextRow == NEXT_ROW)
  {
    /* Transmit new row. */
    ChanOutInt(pFromMux[Index], y);
    if (++y >= NUM_OF_ROWS)
      break;
  }
  else
  {
    /* Otherwise wait until Mand() */
    /*  transmits its buffer.*/
    Index = ProcAltList(pToMux);
    ChanIn(pToMux[Index],
           (void *)LineBuffer, 320);

    SendRow(NextRow);
  }
}

/* Now shutdown all running Mand() */
/*   processes.*/
for (y = 0; y < NumOfProcs; y++)
{
  Index = ProcAltList(pToMux);
  NextRow =  ChanInInt(pToMux[Index]);
  ChanOutInt(pToMux[Index],TERMINATE);
}

getkey();
SetMode(2);
SetPage(0);
exit_terminate(0);
}
```

*Figure 5 - Multiplexor Main() process*

current processes. Unfortunately the more obscure functions (eg the View option to display which process is running on which processor) are not readily available because the manual doesn't go into enough detail on how to use them. If a windowing environment was used then perhaps a comprehensive on-line help facility could be included.

## Conclusion

The B008 mother-board provides an ideal stepping stone into the world of concurrent programming on a PC. Although the concepts behind parallelism are easily understood, in practice, the path to parallel programming is riddled with unforseen complexities. An existing application cannot simply be adapted to take advantage of parallel processing; it would have to be redesigned from the bottom up instead. This does not blend very well with, presumably, the main reason why INMOS decided to supply a C development tool kit instead of its traditional occam system ie to allow existing applications to be ported to the transputer with minimal change. The transputer is ideal for distributing the calculations that are needed for picture processing or ray-tracing as these techniques lend themselves very well to distributed processing. There are several such applications that could be parallelised with relative ease, but even these would require much rethinking since the established algorithms have been optimised for the more conventional, sequential processors.

If problems have been encountered when writing code, INMOS' technical support provides a truly first-rate service. However, I found that this level of support is almost essential as the manuals supplied with both the hardware and the development kit leave much to be desired. The software tools allow applications to be developed and debugged using an extremely comprehensive run-time library and debugging environment. INMOS does not currently supply an integrated development environment and, on a large project, this may slow down the development cycle somewhat. Perhaps a windows driven system, especially for debugging (IDEBUG is presently text-only), could be incorporated into a later release. INMOS' philosophy is that the ISERVER should be bent and shaped by the applications programmer to suit the application, although INMOS is now shipping the B008 with a graphics TRAM module (the Editor didn't tell me this until after I'd written the Mandlebrot thing). If enough applications are written, one day we may well see a supercomputer on our desktops, driven by the rebirth of the transputer.

EXE

*The B008 motherboard costs £658.42. The B404-15 (20 MHz T805) TRAM with 2MB of DRAM and 32KB of SRAM costs £1093.71 and the ANSI C Compiler tool set costs £458.57. Many thanks to Steven Doyle and Nigel Holders at INMOS for their unquestionable patience. The complete B008 tool set is distributed in the UK by Rapid Silicon on 0734 752266. INMOS can be contacted on 0454 616616.*

# pcANYWHERE IV/LAN™


# Don't leave your LAN without it

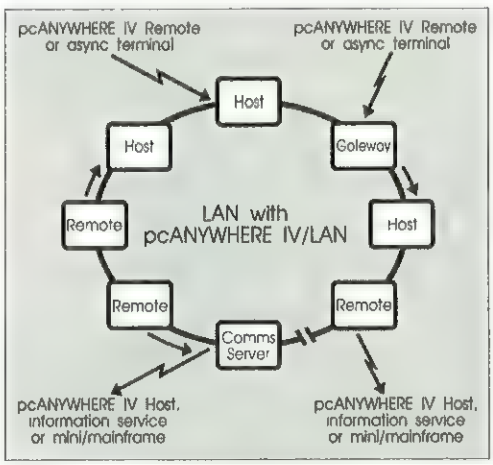
**pcANYWHERE IV/LAN will revolutionise the way your local area network is used. With pcANYWHERE IV/LAN you can remotely control another PC, regardless of whether it's another workstation on the LAN, or on another LAN or a stand-alone PC off-site. From providing technical support to working from home, things may never be the same again. In fact, you may wonder what you did without it.**

pcANYWHERE IV/LAN is perfect for project collaboration, and ideal for end user support and training. Co-workers can dynamically share ideas while working on the same document or spreadsheet. Support technicians can see users' problems and show them the solution right on their PC screens, without ever leaving their desks.

pcANYWHERE IV/LAN further increases your LAN's productivity by providing every workstation with complete general purpose communications functions, including VT220 terminal emulation and bi-directional high speed file transfer.

And, it's powerful gateway function allows any modem attached to a workstation to be used in background by any other network user.

Now, every user can be just a phone call away from direct access to your company's LAN, day or night.




pcANYWHERE IV/LAN is designed to provide maximum flexibility. It supports both NetWare IPX and NetBIOS transports and works with all Int 14 or NASI/NCSI compatible asynchronous communication servers.

For serial communications, full UK modem support means that you can use any of the latest generation of high speed and error-correcting modems.

**For the most complete LAN communications solution, the choice is obvious. Choose pcANYWHERE IV/LAN: For remote computing any way you need it.**

| | |
|---|---|
| pcANYWHERE IV/LAN | £395 |
| pcANYWHERE IV/LAN "Node" | £30 |
| pcANYWHERE IV | £150 |
| pcANYWHERE IV/Host | £110 |
| pcANYWHERE IV/Remote | £85 |

Prices do not include delivery or VAT.

**Telesystems Ltd**

3 WYCOMBE ROAD, PRESTWOOD, BUCKS. HP16 0ND.
Telephone: 02406 6365 Fax: 02406 6050

CIRCLE NO. 147

Authorised DMA distributor.

pcANYWHERE IV/LAN and pcANYWHERE IV are available from all major distributors and dealers. We welcome trade and OEM enquiries.

# Heavy Metal?

*Following our general overview of GUI libraries last month, we will be taking a more in-depth look at individual packages. Joe Borkoles has been testing the Zinc Interface Library V2.0.*

The Zinc Interface Library (henceforth ZIL) is a C++ class library for building CUA-conformant form-based windowing systems. Applications built with ZIL can run under MS-DOS (in both text and graphics modes) and Microsoft Windows, all from a single set of source code. ZIL is currently offered for Borland C++ and Zortech C++, and can use any video standard supported by Borland's BGI or Zortech's Flash Graphics libraries.

ZIL is now in its second major incarnation. There are significant enhancements over the previous version, including an interactive design tool and persistent window objects. Figure 1 shows the V2.0 class hierarchy and summarises the actions of the main interface objects.

## First impressions

ZIL is an impressive package. There are three thickish manuals with 1068 pages between them, and six 720 KB disks - two each for the DOS library, the Windows library and the source code. The disks I received were designated for the Borland C++ V2.0 compiler only, although experiment suggested that the non-Windows components also worked with Turbo C++.

The INSTALL program asks for a base directory for the ZIL, and offers a sensible set of options, allowing partial installations. The ZIL files are held in compressed form on the distribution disks, and are automatically expanded by INSTALL. The complete installed system occupied just over 3 MB on my hard disk.

Installation creates a directory structure containing subdirectories for LIB, BIN, INCLUDE, SOURCE, EXAMPLES and TUTOR. LIB contains a further subdirectory, WINDOWS, for the Windows libraries. There are three versions of the library, ZILL.LIB (no persistent objects), L_ZILL.LIB (loading of persistent objects) and S_ZILL.LIB (loading and saving). BIN contains two tool programs: GENHELP.EXE, which converts plain text files into help files, and DESIGN.EXE, an interactive screen designer. The INCLUDE directory is for header files containing the class definitions and globals. The SOURCE directory is self explanatory.

There are 28 zipped files (but no unzipper) in the EXAMPLES directory. These give a variety of example programs, some of which are very useful, including a check box class, radio box class, file editor, calculator, and digital and analogue clocks. The TUTOR directory contains source for the various tutorials, including the inevitable 'Hello world' and an extensive example which

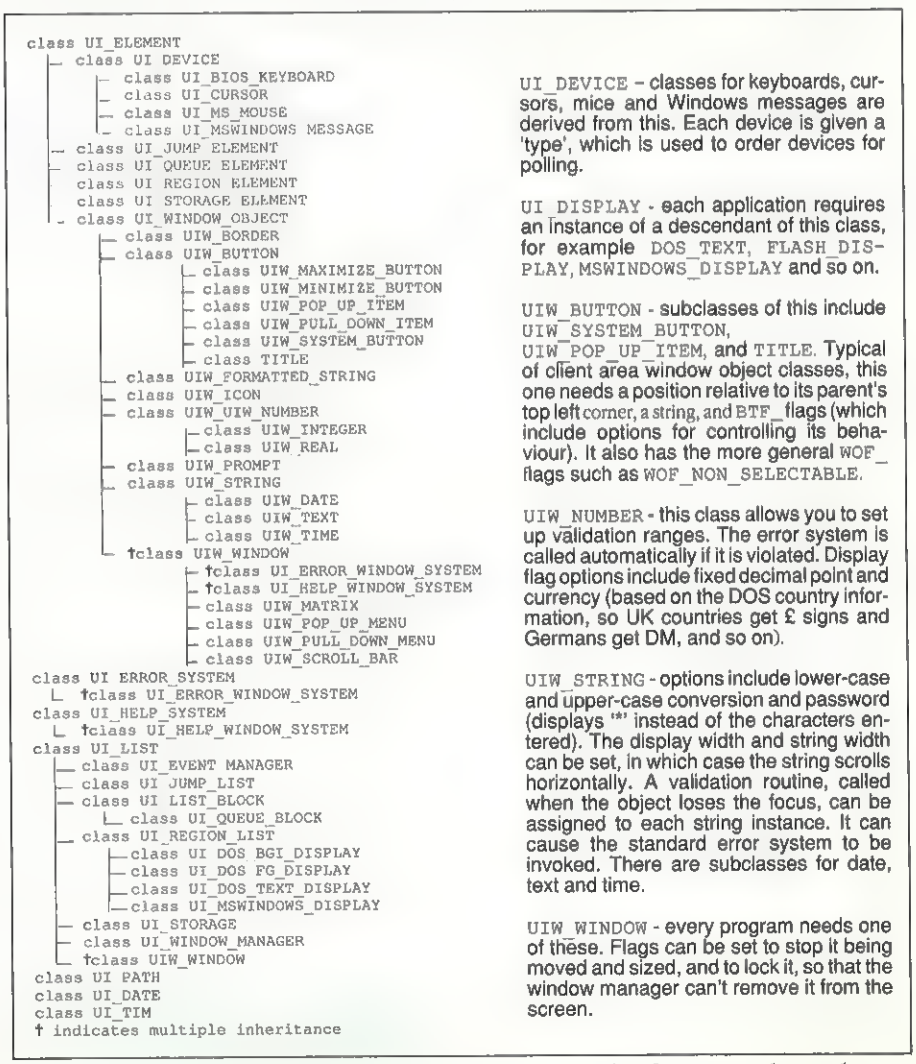



*Figure 1 - ZIL hierarchy, with some example objects explained*

shows off most of the functionality of the library. A 5.5 KB text file lists the changes from V1.0. The READ.ME file encourages people to register, with the incentive of receiving a Windows version of the screen designer program when it becomes available.

The three manuals are a *Programmers Guide*, a *Tutorial* and a *Reference*. There is also an A5 quick reference card. The documentation is generally of high quality, and addresses the material at different levels appropriate to the skill of the user. The novice ZIL user is gently introduced to the overall concerns and conceptual design issues of the product. There is a good 'hands on' approach in the tutorial, and finally a work-horse reference volume. The effort applied to these manuals is highly commendable.

## The main( ) Event

How does ZIL manage to provide sophisticated Windows, graphics and text based programs from a single set of source code? The core of the system is an event loop, which waits for devices to generate events, then dispatches them to the window manager. This design permits quite straightforward source code - see Figure 4, which lists a typical 'main' file, containing `main()` and `WinMain()` functions. All the different programs I have developed with ZIL were based on the same main file - only the name of the class used as the controlling window needed to be changed.

Figure 4 also shows all differences between Windows and non-Windows code required in any ZIL program. ZIL header files require that `ZIL_WINDOWS` is defined to create a Windows application. The code here uses `#ifndef` to exploit this convention, allowing both non-Windows and Windows stuff to live in the same file. As you can see, Windows requires a `WinMain()` instead of a `main()` function, plus different display and event manager classes. However, every other piece of code is identical between the two environments.

The first step in both `main()` and `WinMain()` is to initialise the display. The vanilla DOS version attempts to set the computer into graphics mode (by instantiating `UI_DOS_BGI_DISPLAY`); if this fails, it uses text mode (which can be 43/50 lines on an EGA/VGA machine). (Incidentally, I was using the Borland version of ZIL, which uses the standard Borland graphics library to access the screen. Zinc claims that the graphics is 250% faster using the Zortech library, which may influence your choice of compiler.) The equivalent Windows code has simply to create an instance of `UI_MSWINDOWS_DISPLAY`.

Next, we need an event manager. Note how, in both sections of code, the event manager is told from which devices it should expect events by use of an overloaded + operator. If you need to support additional devices, they can easily be created by inheriting from the base class `UI_DEVICE`. This is a good example of the versatile design of ZIL's class hierarchy - extensibility is not only possible, but seems positively inviting!

At this point we come to the environment-independent code. First, the display and event manager instances are passed to the window manager constructor. We are now ready to add an application-specific window (again using an overloaded + operator). The example shows how to add a control panel object of type `ReferencesView`. The error and (non-Windows Help compatible) help systems are cranked up.

The last few lines of code contain a `Get/Put` event loop and tidying up. The event loop waits for events from the event manager and feeds them to the window manager. It terminates when there are no windows to receive messages, indicated by the window manager's returning `S_NO_OBJECT`. On termination of the loop, the destructors must be called in the proper order.

There is one minor complication to this otherwise elegant scheme. It is possible to switch


*Figure 2 - An application running in DOS graphics mode*

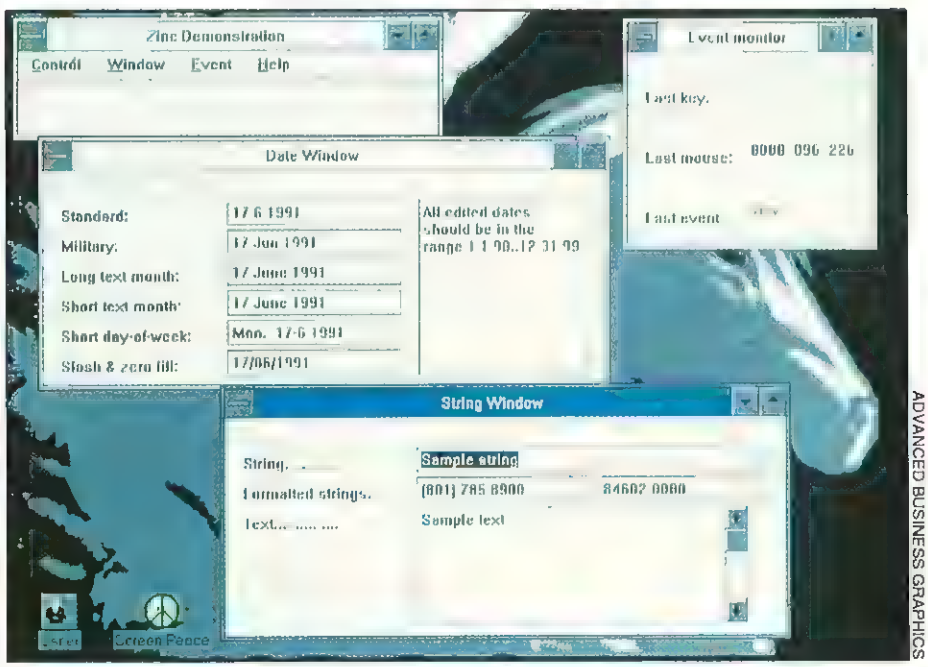



*Figure 3 - Same application under Windows*

between text and graphics DOS modes on the fly. However, this can only be done at the top level in `main()`, by modifying the event loop to look out for the appropriate message.

## Persistence

Figure 5 shows two alternative constructor declarations for a simple control panel window, subclassed from `UIW_WINDOW`, containing a single menu option, a prompt and a string field. The first uses only ZILL.LIB (no persistent objects). A constructor call is made to the `UIW_WINDOW` constructor with the window's size parameter, then all the internal components are Added into the window. The second version of the constructor uses L_ZILL.LIB (or S_ZILL.LIB) to load the window object from disk (again, via a call to `UIW_WINDOW`'s constructor). This version is obviously preferable for complex windows where an ordinary constructor would be very large. In either case, this is all the code that is necessary to cause the window's components to be displayed - refreshes will take place automatically.

When subclassing a window in this way, one usually needs to override the `Event(UI_EVENT &event)` method. This method receives UI_EVENT instances dispatched by the window manager to the top window. These are generally intercepted by the application's window class so that the appropriate action may be taken. For example, it is possible to arrange it so that menus send programmer-defined events to the top window. Events not recognised by application-specific code are dispatched to the parent by calling the `UIW_WINDOW::Event()` routine.

There is not space in this article to describe the use of the library in more detail. However, the general approach will be very familiar to those who have used other event-based application frameworks, such as Borland's Turbo Vision.

## The Designer

There is an important link between persistent object support and the Screen Designer. A persistent window object contains all the attributes of a window - size, position, field, field contents and so on - saved on disk. When reloaded, the window will appear as before. This technology allowed Zinc to create an interactive design tool which saves screen designs to disk for retrieval when necessary.

The Screen Designer allows you to create any of the standard window objects (not, unfortunately, the 'extra' classes such as check boxes and radio buttons) and set up their initial contents, range validation procedures and option flags. So far, so good. But once I started to use the Designer in earnest, its benefits evaporated rapidly. I have lost several hours of work because of bugs in the software. The faults are horrendous, though, in mitigation, I can report that I have discovered work-arounds for about 95% of them.

Here are some examples. The program will happily let you quit without telling you if you have saved the last changes. The editor allows you to clear, cut, copy, paste and delete items... but there is no indication of what the currently selected item is. The only way to find out is to double click it and invoke its options editor. It is also possible to create a file with the wrong extension, which the system pretends to save, but doesn't. It is possible to overwrite work if you create a new file with the same name, without any kind of warning. Prolonged use of the editor inevitably ends in a crash. I could not get the designer to work with Windows applications at all!

I was looking forward to this Screen Designer more than any other thing in the package. Perhaps that's why I am so bitter about it. For a company that is producing a software engineering tool kit for the 1990s, it really is an appalling piece of software. I have contacted the support staff at Zinc about my problems. They promised to send me a better version. At the time of writing, this has not appeared.

Is the program worth using? Well, I have worked around most of its problems, and I do now use it extensively. I suppose that is the real test. I can only hope that Zinc will forward copies of a working version of the editor to *all* users without cost in due course. The Zinc Interface Library does sustain a very high standard for every other item in the tool set: its manuals are large, well-written and attractively printed, the classes provided are extensive and cleverly designed, and it is a pleasure to use. If they could only get the Screen Designer working as well...

What else is wrong with ZIL? This is really more of a wish list than a list of problems. I would like better support for the Windows help style, perhaps even extending support to the DOS version. Some support for printers in the DOS mode might be nice. Perhaps my single biggest wish is for a much more sophisticated `UIW_TEXT` object, which could cope with multiple fonts, searching etc.

## Source

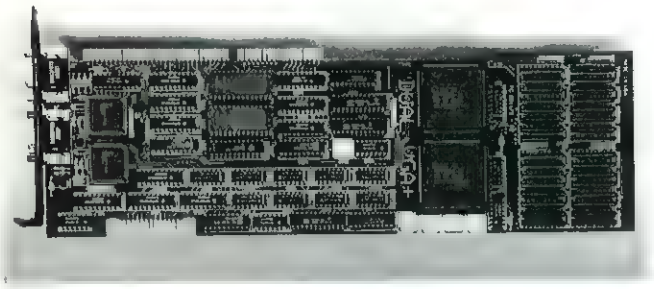
The source code is split into two parts. The DOS source consists of 118 files in all, of which 39 are for the Screen Designer. The Windows subdirectory contains 80 files - the difference is explained by the fact that there is no Windows version of the designer. The Windows directory also contains a zipped file of pre-compiled .OBJ files for creating a DLL. The code is beauti-

```
#ifndef ZIL_WINDOW

main ()
{
  // Initialise the display,
  // trying for graphics first
  UI_DISPLAY *display;
  display = new UI_DOS_BGI_DISPLAY;
  if (!display->installed)
  {
    delete display;
    display = new UI_DOS_TEXT_DISPLAY;
  }

  // Initialise the event manager and
  // add three devices to it.
  UI_EVENT_MANAGER *eventManager;
  eventManager =
      new UI_EVENT_MANAGER(1000, display);
  *eventManager
    + new UI_BIOS_KEYBOARD
    + new UI_MS_MOUSE
    + new UI_CURSOR;

#else

int PASCAL WinMain( HANDLE hInstance,
                    HANDLE hPrevInstance,
                    LPSTR lpszCmdLine,
                    int nCmdShow )
{
  // Initialise the display
  UI_DISPLAY *display =
    new UI_MSWINDOWS_DISPLAY(hInstance,
                             hPrevInstance,
                             nCmdShow);
  if (!display->installed)
  {
    delete display;
    return 1;
  }

  // Initialise the event manager
  eventManager =
        new UI_EVENT_MANAGER(100, display);
  *eventManager + new UI_MSWINDOWS_MESSAGE;

#endif

  // Initialise the window manager.
  UI_WINDOW_MANAGER *windowManager;
  windowManager =
    new UI_WINDOW_MANAGER(display,
                          eventManager);
  *windowManager + new ReferencesView;
  // Initialise the help and error systems.
  _errorSystem = new UI_ERROR_WINDOW_SYSTEM;
  _helpSystem =
    new UI_HELP_WINDOW_SYSTEM("helpfile",
                              windowManager,
                    GENERAL_HELP);

  // Wait for user response.
  int ccode;
  do
  {
    UI_EVENT event;
    eventManager->Get(event, Q_NORMAL);
    ccode = windowManager->Event(event);
  }
  while (ccode != L_EXIT &&
                  ccode != S_NO_OBJECT);

  // Clean up.
  delete _helpSystem;
  delete _errorSystem;
  delete windowManager;
  delete eventManager;
  delete display;
}
```

*Figure 4 - Typical main/WinMain of ZIL application*

MULTIPLE INDEPENDENT MONITORS
CAN NOW BE DRIVEN SIMULTANEOUSLY
FROM A SINGLE PC OR PS/2

# THE DUAL VGA PLUS



★ UP TO 800×600 RESOLUTION
★ TWO INDEPENDENT SIMULTANEOUS DISPLAYS WITH OVERLAY CAPABILITY
★ 100% COMPATIBLE WITH IBM VGA/EGA/CGA/MDA VIDEO STANDARDS
★ 4 BOARDS IN ONE SYSTEM GIVE 8 SIMULTANEOUS DISPLAYS

**NEW DRIVER! ALLOWS WINDOWS 3.0 TO BE EXPANDED ACROSS FOUR MONITORS**

**CEBRA COMMUNICATIONS LIMITED**
26 LORNE PARK ROAD, BOURNMOUTH
TEL 0202 299048 FAX 0202 299192

CIRCLE NO. 154

# 262,144 colours in text mode for Clipper, FoxPro and Clarion developers

Tailormade Limited, a leading developer of bespoke systems announce the launch of the VGA colour palette toolbox.

- Allows you to assign any one of the 262,144 colours to the normal 16 colours available in VGA text mode.
- Results in a distinctive and highly individualised user interface.
- TSR pop-up palette changer for testing colour combinations during development.
- Easy integration into your applications, with minimal memory overheads.
- £69 plus VAT, postage and packing.

Tailormade Ltd, Tudor House, 649 London Road, High Wycombe, Bucks HP11 1EZ. Tel: 0494 473338 FAX: 0494 473350

CIRCLE NO. 155

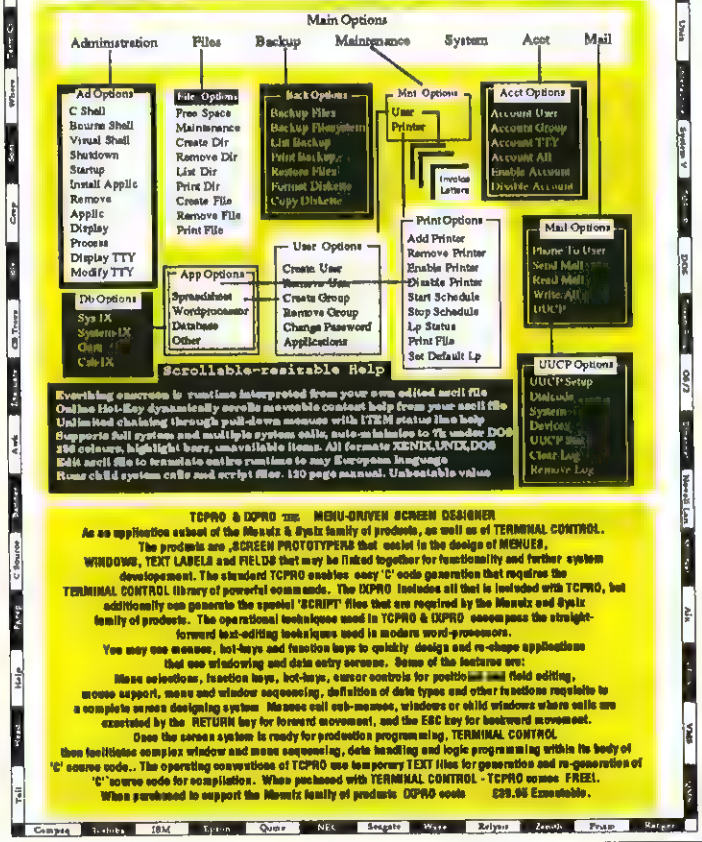
Probably THE best deals in the entire United Kingdom !.

Xvale Ltd., Store 'D'
55, Bedford Court Mansions,
Bedford Avenue, London.
WC1B 5AD U.K.

**TEL: 071-636-8210 / 071-631-4818**
**FAX: 071-255-1038 24 Hours**

Barclaycard
Access

**MENUIX and SYSIX £49.95 - £99.95**




Everything concrete is runtime interpreted from your own edited ascii file
Online Hot-Key dynamically scrolls moveable context help from your ascii file
Unlimited chaining through pull-down menuus with ITEM status line help
Supports full system and multiple system calls, auto-minimizes to fit under DOS
256 colours, highlight bars, unavailable items. All formats XENIX,UNIX,DOS
Edit ascii file to translate entire runtime to any European language
Runs child system calls and script files. 120 page manual. Unbeatable value

**TCPRO & DCPRO THE MENU-DRIVEN SCREEN DESIGNER**

As an application subset of the Menuix & Sysix family of products, as well as of TERMINAL CONTROL.
The products are ,SCREEN PROTOTYPERS that assist in the design of MENUES, WINDOWS, TEXT LABELS and FIELDS that may be linked together for functionality and further system development. The standard TCPRO enables easy 'C' code generation that requires the TERMINAL CONTROL library of powerful commands. The DCPRO includes all that is included with TCPRO, but additionally can generate the special 'SCRIPT' files that are required by the Menuix and Sysix family of products. The operational techniques used in TCPRO & DCPRO encompass the straight-forward text-editing techniques used in modern word-processors.
You may use menues, hot-keys and function keys to quickly design and re-shape applications that use windowing and data entry screens. Some of the features are:
Menu selections, function keys, hot-keys, cursor controls for position and field editing, mouse support, menu and window sequencing, definition of data types and other functions requisite to a complete screen designing system. Menues call sub-menues, windows or child windows where calls are executed by the RETURN key for forward movement, and the ESC key for backward movement.
Once the screen system is ready for production programming, TERMINAL CONTROL then facilitates complex window and menu sequencing, data handling and logic programming within the body of 'C' source code.. The operating conventions of TCPRO use temporary TEXT files for generation and re-generation of 'C' source code for compilation. When purchased with TERMINAL CONTROL - TCPRO comes FREE!.
When purchased to support the Menuix family of products DCPRO costs £99.95 Executable.

Probably THE best deals in the entire United Kingdom !.

Xvale Ltd., Store 'D'
55, Bedford Court Mansions,
Bedford Avenue, London.
WC1B 5AD U.K.

**TEL: 071-636-8210 / 071-631-4818**
**FAX: 071-255-1038 24 Hours**

Barclaycard
Access

# UNIX – & – LNX

**LNX Notebook - 20 (386-16/20 m'board)**
**4 MB ram, 3-1/2 Drive, 120 MB hard-disk**
**VGA 640 * 480, 32 grey scales**
**1 Ser / 1 Par / 1 Vga, 77 keyboard**
**UNIX system V / rel 3.2 / v2.02**
**Network Platform £2495** *TRAVELLER*

**LNX system 38, (386-33 mhz m'board)**
**8 MB ram, 5-1/4 & 3-1/2 drives**
**SVGA 1024 * 768 screen, SVGA adapter 1MB**
**2 Ser / 1 Par / 1G, 102 Keyboard, Mouse**
**300 MB hard-disk, Mini-Tower,**
**UNIX sys V / rel 3.2 / v2.02**
**Workstation Developer £2650** *BUSINESS*

**LNX system 48, (486-33 mhz m'board)**
**16 MB ram, 5-1/4 & 3-1/2 drives**
**SVGA 1024 * 768 scrn, SVGA adapter 1 MB**
**2 Ser / 1 Par / 1G, 102 Keyboard, Mouse,**
**600 MB hard-disk, Full or Mini-Tower**
**UNIX sys V / rel 3.2 / v2.02**
**Workstation Developer £3995** *CORPORATE*

CIRCLE NO. 150

fully laid out. The true object oriented design of ZIL greatly helps when trying to understand it. The comments, however, are a little sparse for my taste.

## Windows?

As has been shown, support for Microsoft Windows is fairly transparent. The addition of a few extra lines (as shown in Figure 4) allows Zinc users to recompile their code to run under Windows 3.0. All the objects behave in the same way as they do in graphics mode. There are, however, a couple of glitches. The Windows client area refresh can get confused, but usually sorts itself out in the end. I have crashed the Windows version of the demo program, which is not really a good sign.

ZIL does not use the native Windows facilities for menus, check boxes and radio buttons. Instead, it creates them from scratch, much as it does in DOS. There may be a minor performance penalty for this approach, but I didn't notice it. Should a developer who wants to create Windows-only programs use the Zinc class library? ZIL reduces the amount of code for any program dramatically, but there are limitations. There is no support for many Windows user interface properties, eg MDI windows and GDI functions. If a program design demanded these things then some tweaking would be required. However, the base classes of Zinc are so well designed that I suppose that this would require a minimum of effort.

## Conclusion

Despite my disappointment with the Screen Designer, I feel that ZIL provides very tangible benefits for developers. I estimate that it reduces the development effort of supporting both DOS and Windows by an order of magnitude. In addition, it greatly simplifies programming in a windowing environment. The beautifully designed class structure, the comprehensive manuals and the availability of the source code, all add up to a premier development tool.

EXE

```
//
// constructor using the manual method
// for adding window contents
//
ControlPanel::ControlPanel() :
                    UIW_WINDOW(1, 1, 30, 15)
{
  // add the prompt and string objects
  Add(new UIW_PROMPT(1, 1, "Status"));
  Add(new UIW_STRING(1, 8, 30));

  // add a pull down menu
  UIW_PULL_DOWN_MENU *menu;
  Add(menu = new UIW_PULL_DOWN_MENU());

  // add a pop up item to the menu
  // and give it a name
  UIW_POP_UP_ITEM *item;
  item = new UIW_POP_UP_ITEM();
  item->StringID("EXIT_OPTION");
  *menu + item;

}

//
// constructor that uses a persistent ob-
ject
// created by the design tool
//
ControlPanel::ControlPanel() :
      UIW_WINDOW("CNTRLRES~CONTROL_PANEL")
{
}
```

*Figure 6 - Alternative constructors*

*Joe Borkoles is a research scientist at Queen Mary and Westfield College. Since 1988 he has been actively involved with interactive systems and object oriented technologies.*

*The Zinc Interface Library V2.0 is available from Zortech in the UK, priced £149.95 (both Borland and Zortech version) including source and Windows support. Zortech is on 081 316 7777.*

# The compiler bundle

*Jules May reckons compiler vendors are exploiting him.*

Let me tell you a story. I wanted to hang a picture the other day, and I couldn't find my hammer, so I went to a DIY shop. The assistant came over, wringing his hands, asking 'What can I do for you?' I told him what I wanted, and he went off to find something.

'Look at this' he said when he came back, 'A 250-piece toolkit - just what you want!'.

'No,' I explained, 'I want a hammer.'

'Its got a hammer, Sir. In fact it's got three. Its also got socket sets in imperial and metric, more screwdrivers than you can...'

'No, I want a hammer. Not three, just one.'

After more such banter, he reluctantly went off again, and returned with a smaller box.

'I've found this. Not as good as the other one, but it's only got one hammer in it. How's that?'

'OK', I said, 'How much?'

'£97.50' he said, 'although I can give you a good discount!'

'How much for the hammer?'

'I can't split the set up, Sir. What are we going to put in the little plastic blister?'

And so it went on.

Or rather, it didn't. I've made this story up to show the stupidity of the situation which I am grumbling about this month. I'm grumbling about compilers.

Compilers are a subject very close to my heart. I like using them, I like writing them, I even use them to write them. I do this because I think that by writing a compiler one can achieve greater efficiency and productivity than any other way.

In the olden days, an apprentice craftsman would learn how to make his own tools. Many of those tools he would have to throw away, because they were useless. Eventually, he would take a test - he would make something with his tools; if he completed it satisfactorily he would be considered a master craftsman. It often took many years to achieve this level of proficiency. This is where the irony about a bad worker blaming his tools came from - the tools were tools he would have made for himself, so his craftsmanship was still poor no matter where he laid the blame.

In order to make good tools, or even use tools effectively, one must have a feeling for the material one is working with, whether it be wood, iron, or computers. Few people make their own programming tools nowadays, and fewer buy specialised, targeted tools.

A case in point is the C preprocessor. What a wonderful piece of software that is! You can set flags inside or outside the file, and control what information is passed onto the next stage. Even in languages which have proper constants, CPP is a Godsend in version control, diagnostics, and multiple models. The selfsame program can be used to keep manuals in synch with the software they are supposed to describe by pre-processing the text before it is formatted.

Where does one get this magnificent program? It's no longer a program in its own right (the way it used to be) with a rich and developing syntax; it's now part of the C specification, and I know of not one publisher who offers it as a separate product from their C compilers - indeed, few C compilers even put it in a separate file - it's part of the C parser, where it can't spread its wings. That strikes me as a great loss.

But even the compilers themselves are built like the DIY shop's tool-boxes. It seems a publisher can't gain any street-cred without offering an integrated development environment, containing editors, debuggers, and graphical error reporting. They offer such things because (presumably) their output and linkage is incompatible with everyone else's. Some publishers, realising they are making their customers' lives hard, offer things like Brief-compatible keystrokes on their editors, not realising that nobody buys Brief because of its user-friendly keyboard language!

'I want to buy a C compiler.' (I don't, but let's play the game.)

'Certainly Sir. Here we have Wobbly C version 23.67a. It comes with its own editor (with Cyrillic option), a debugger that won't debug anything except Wobbly, and a Teas-maid, all integrated seamlessly onto a lawn-mower..'

The problem is that when you open the box, install another 5 MB of software which duplicates the functions of the other 30 MB of integrated environments already on the disk, and actually throw some code at it, it hardly ever works properly. The editors fall over, the environment variables interfere with each other, and the compilers give up because they've run out of memory. At the end of the day, if you've got a decent editor anywhere on the disk, you will want to use it for pretty much everything. If you have a decent debugger, you will want to work with it until it fits your hand like an old friend. If you can't, because it is incompatible you will want to use another compiler which is compatible.

Hence this appeal for modular compilers. If a publisher thinks he can write decent compilers, that's fine. If he thinks he can write decent editors as well, that's fine too, but he should consider that I may not share his jack-of-all-trades image of himself. I may like his compiler and hate his editor. I don't want integrated environments, I want compilers in bits - I want code-generators which I can recycle to other languages, I want optimisers that I can plug anything into, and I want to buy them separately.

EXE

---

*Jules May is still a consultant specialising in graphics and HCI. In the course of his business he is moved to write six or seven compilers per year. He may be contacted on CIX as* `Jules`, or on 0707 44185.

# .EXE directory

This is a selection of products from the .EXE directory. Call NOW for a complete catalogue, or to place your order direct.
0442 824501

## .EXE Disks

*A collection of specially selected source listings from .EXE, and a selection of Public Domain and Shareware utilities. Each disk contains around 1.4 Mb of data and an average of 15 files per disk. Some of the contents are given below.*

**Volume 4 - ref: ED45 - 5.25" or ED43 for 3.5" disks**
BOSS A superb text windowing package for C. It lets you create, write to, move and generally manipulate text windows. It also lets you create pull down and pop-up menus.

**Volume 5 - ref: ED55 - 5.25" or ED53 for 3.5" disks**
CTASK20.LZH CTASK V2.0 is a multi tasking kernel for C. It's a complete task schedule for MS-DOS that allows you to run parts of your C program in parallel. MS-DOS call can also be made concurrently.

SNAP402.LZH Version 4.02 of SNAP, an excellent dBASE code documentor which now handles dBASE IV and FoxPro too.

**Volume 6 - ref: ED65 - 5.25" or ED63 for 3.5" disks**
INTER290.LZH Source code, in C and assembler, to LHARC, the file compression/decompression program as used on many of the .EXE disks.

SCANV64.LZH A virus scanner which will scan your hard disk and search for around 50 known viruses.

**Volume 7 - ref: ED75 - 5.25" or ED73 for 3.5" disks**
CDRIVER.LZH A collection of files that explain how to write MS-DOS device drivers in C.

4DOS301.LZH 4DOS is a complete replacement for COMMAND.COM. 4DOS ads a number of new features, including command aliases, command recall and an extended DOS batch language.

**Volume 8 - ref: ED85 - 5.25" or ED83 for 3.5" disks**
LHARC- compression software, for packing & unpacking programs. BACKMENU- throws up a pop-up menu with selection of launchable programs whenever desktop is clicked. CV- CodeView for Windows. Complete source & assembly level Windows debugger. DDEWATCH- debugs DDE programs. Lists, translates, & logs all DDE calls while in background. WINPOST- Post-it notes for Windows. WINSTUB- better DOS stub program. Doesn't print error message when Windows programs run from DOS, automatically executes Windows, & invisibly re-runs application.

**Volume 9 - ref: ED95 - 5.25" or ED93 for 3.5" disks**
LHARC- See Volume 8. BACKISS- Back issue list for .EXE. From Vol 1, No 1 to Vol 5 No 9. DPMI- DOS Protected Mode Interface GNURCS- GNURCS is a DOS port of GNU's Unix RCS. SID- DES encryption utilities. Source & executables for entire DES implementation. SVGA- library for 256 colour Super VGA adaptors.

**Volume 10 - ref: ED105 - 5.25" or ED103 for 3.5" disks**
LHA- version 2.1 - public domain compression program (was LHARC). One of best speed & compression ratios of any compaction program. LHSRC- full source to LHA program. BOOKLET- shareware program, similiar to PR2, allows 80-column manuals reprinted as A5 booklet.

BOYMOORE- fast search of strings. Full implementation of Boyer-Moore algorithm. DOSTRACE- a TSR utility for 386 machines monitors & lists interrupt calls, open files, watches addresses, dumps memory, breaks on software interrupts. INFOPLUS- relays information on your PC. Turbo Pascal source shows how it works. INTER191- Famous Interrupt List. MOUSEQ- queues mouse interrupts. Full C source. POPBUG- newly discovered 386 bug, affects POPAD instruction. Test code & description. PR2- ANSI C program, prints 80-column text files in 2 columns on certain Epson compatibles. SPAWNO30- Replacement spawn function for MS & Turbo C swaps a running program to disk, EMS, or XMS memory while executing a child program or subshell. STATEMAC- Ray Jones' state machine interpreter in C.

## .EXE Library

SPECIAL OFFER

*Even the best programmer needs a good set of reference books by his side. We have bundled a few books with collections of bound articles from .EXE. substantial savings can be made by ordering the two. All books are available separately. Please call 0442 824501 for individual prices.*

**Intensive C Course Book** - illustrates the principles of good programming style and sound design. Available @ £10 with the collection of articles on C from .EXE. (Would normally be £15.20). ref: **EBAC**

**Liskins dBASE IV Programming** - gives plenty of applications to demonstrate the elements of dBASE from fundamentals to applications design. Available @ £28.95 including collection of articles on dBASE (normally £36.45). ref: **EBAD**

**The Master C Book & Disks** - A comprehensive, High Powered book and software package which revolutionises the instruction of C. 4 disks are provided which automatically guide you through C topics. An excellent way to learn. **£39.95** ref: **EBMC.**

**Professional Programmers Guide to Pascal** - tutorial and reference material are carefully integrated. Standard and Extended Pascal are both covered in detail. Available @ £11.00 including a collection of articles from .EXE on Pascal programming. (Normally £14.49). ref: **EBAP.**

**The Professional Programmers Guides** adopt a no-nonsense approach to programming and are designed for the serious programmer. Ideal for the novice or as an aide memoir for a\the professional programmer.

**Fortran 77 - £8.99 ref: EBKF** **Ada - £9.99 ref: EBKA**
**Modula-2 - £8.50 ref: EBKM** **Pascal - £8.99 ref: EBKP**

## .EXE Mugs

Stylish white mugs with a green .EXE logo - guaranteed to be of use every day in the office! **£4.50** ref: **EM1W**

## .EXE Car Sticker

White oval car sticker (like the GB stickers used when driving abroad) printed with the .EXE logo - only other .EXE readers will know where you're really coming from! **£1.50** each. ref: **ECS1**

## .EXE T-Shirts

.EXE T-Shirts, featuring the .EXE Logo or one of 4 other slogans will make you the envy of the office. Available in either white or dark green in Large **£9.00** or Extra Large **£9.50**

Reference number - please quote when ordering:

| | *White* | | *Green* | |
|---|---|---|---|---|
| *LOGO* | *Large* | *X Large* | *Large* | *X Large* |
| *Born to compile* | *ET1WL* | *ET1WXL* | *ET1GL* | *ET1GXL* |
| *.EXE Developers do it best* | *ET2WL* | *ET2WXL* | *ET2GL* | *ET2GXL* |
| *'As he returned from the computer room' (cartoon from Oct '89 issue of .EXE)* | *ET3WL* | *ET3WXL* | *ET3GL* | *ET3GXL* |
| *'Verity Stob has NOT got a big bottom'* | *ET4WL* | *ET4WXL* | *ET4GL* | *ET4GXL* |
| *.EXE Logo* | *ET5WL* | *ET5WXL* | *ET5GL* | *ET5GXL* |

## ORDER FORM

*I would like to order the following products from the .EXE directory:*

DESCRIPTION REF. NUMBER QTY UNIT PRICE TOTAL PRICE

Postage & Packing £1.00 TOTALPRICE

My total order value is £ _____ *(All price include VAT where applicable)*

*I would like to pay as indicated (please tick):* ❑ Cheque enclosed

❑ Please debit my VISA/Access card No:

Card No: _____ Expiry: _____

Name: _____

Company: _____

Address: _____

_____ Postcode: _____ Tel: _____

Signed: _____ Date: _____

***Please return to Process Communications Ltd, PO Box 388, Tring, Herts HP23 4EB.***
***Please quote reference numbers when ordering goods.***

# PPL - The P-Stat Programming Language

*Statistics packages are a neglected source of programming power. Peter Flynn puts one of the big ones through the hoops.*

More years ago than I care to remember I was assigned to a project where the data had been collected manually, on scraps of paper, backs of envelopes, sweet bags, the back of exercise books; almost everything except cigarette packets. Naturally, none of it was in the same format, and almost all of it was mutually incompatible in its raw form. You've all seen this kind of thing.

Having arranged for the data to be transcribed and punched (yes, punched, that says how long ago it was), I was handed a manual for the statistics package to be used. I was about to make the point that software selection would normally be my decision, when, leafing through the book, I spotted some interesting statements - programming statements.

The software was called P-Stat, and I've been using it on and off ever since. Naturally it has been significantly updated over the years, but it has remained one of the Big Four (the others being the better-known SAS, SPSS and BMDP). Originally an IBM mainframe package, it has now been ported to PCs, UNIX and most other platforms (except the Mac).

***PPL is one of the shorter Triangle Problem solutions***

The feature that attracted me was the P-Stat Programming Language, PPL. Lots of stats packages have had their own programming statements for a long while, but the syntax of this one was substantially easier than most, and could be tacked onto any of the statistical commands to provide on-the-fly manipulation of the data as it was passed to the stats routines. This made the task of massaging the data into usable form very much easier, and I now keep a copy as part of my toolkit for doing exactly this kind of job, which crops up more often in my line of business than statistical work does.


## File processing

P-Stat maintains data in a proprietary file format. The data can come from disk, tape, CD-ROM or whatever, or can be typed in at the terminal. The `BUILD` command (which creates P-Stat files) lets you name variables, specify validity checks and cater for missing, corrupt, or spurious data. It will read almost anything: fixed-format (columnar data); free-format (delimited, with alphabetic fields in quotes); and binary (word-aligned) data, all with single or multiple physical records per logical 'case', including groups with missing records. It will also read and write files from other systems such as SAS, SPSS, BMDP, dBASE and Informix, and from DIF files.

Once you have created a P-Stat file, the usual range of statistical procedures is available. These are fairly extensive, as you would expect from a large stats package, and cover pretty much everything I've ever needed in the way of analysis. Also included are many data management commands which provide an environment approaching that of a relational DBMS. You can sort, merge, interleave, separate, compare, concatenate, join, aggregate, collate and look-up files in almost any way you like, which is what makes it useful for massaging data.

## Programming

The construct for PPL in a command is to follow the filename by program statements in brackets. These statements get tokenised (semi-compiled) and processed as the file

```
hist animals[if live.weight good, retain;
     gen log.feed to log(last.week.feed);
     set age to recode(age,6 to 12=1,13 to 24=2,
         25 to 36=3, 37 to 48=4, x=5);
     keep live.weight log.feed age]$
```

*Figure 1 - Example of PPL processing in a statistical command*

is read, before the data is handed to whatever stats or data management routine you are specifying. There is also a PROCESS (no-op) command to let you just use the PPL on a file without going near any of the stats stuff.

Figure 1 contains an example of some processing applied to a file of animal measurements before producing histograms. Only those records with a valid live weight measurement are required, a new variable log.feed is generated, equal to the log of last week's feed intake, and age (which is in months) is re-coded into age-bands numbered 1-5. Finally, only the three named variables are kept for histogramming: but the original data in the file is left untouched - unless an output filename is given, PPL modifications apply only to the current command.

## Command syntax

Commands like the one in Figure 1 are frequently multi-line, so the dollar-sign is used to signal the command end (on other lines you just hit Return). In the PC and some other versions, there are menus and windows, so that you can build a command without all the typing. As with most such systems, once you get used to the syntax, it's actually faster to bash in a command line than faff around with menus. The syntax is straightforward: command name, then file name (with optional PPL in brackets), followed by any identifier clauses with their arguments. For example, the LIST command can take many formatting identifiers, such as MARGIN 10 or LINES 70, to control appearance.

## PPL facilities

PPL itself can be used to select cases and variables for processing (explicitly or by wildcarding multiple names) and can change the order of the variables (from left to right across the case). You can change the value of variables, delete them, generate new ones, and it has the usual range of arithmetic and character functions you expect in a programming language, plus a lot of logical tests (see below).

Repetitive processing across the variables in a record is performed with a loop, by specifying a counter and the start and finish variables: given a file with (amongst other things) twelve contiguous monthly measurements called Jan.Weight to Dec.Weight, you could say

***I now keep a copy as part of my toolkit for doing exactly this kind of data massaging***

```
[do #j using jan.weight
 to dec.weight;
if v(#j) missing,
 set v(#j) to 0;
enddo]
```

where v(#j) is the jth variable in the domain of the loop. Equally valid would be

```
[do #j using ?.weight;
if v(#j) missing,
 set v(#j) to 0;
enddo]
```

if the variables were unordered, or interspersed with other variables (non-contiguous). If it is not necessary to identify individual variables, but still check for a condition, the even faster

```
if any(?.weight) missing,
put "Cannot handle pig" (.n.)
"due to missing weight data",
exclude]
```

which would display the error message on the screen and omit the case from the current command. DO loops can also extend over several lines of PPL, each ending with a semicolon, with control provided by NEXTDO and EXITDO, and a target label marking the endpoint of the loop. IF...THEN...ELSE...ENDIF blocks are also supported for complex conditions.

The MODIFY command, to which you can attach arbitrary PPL statements, simply outputs the modified data to another file (see Figure 2: this would make a file of animals sent for slaughter, with their reference number, weight measurements and date being the only variables kept).

You can split a single case into multiple cases, with inheritance of specified variables, and collapse multiple cases into a single case. This, combined with the commands for matching files on common variables, provides an excellent means of managing hierarchical data structures in a relational manner.

The LIST command prints aligned columns, with variable names heading them, correctly propagated from page to page, with headers, footers, page numbers and so on, very similar to the better report-writing packages. There's a TEXT.WRITER command as well, which formats text reports like a word processor.

## Sundries

One drawback on PC and mainframe versions is the absence of graphics, apart from terminal plots and histograms done with Xs and *s. If you are running X-Windows, however, the graphics are superb. My main use is for the data handling: if I want business graphics from data on the PC or VAX, I squirt it out to disk in a form that another package can read.

What is nice though, is the recent addition of a PostScript option for cross-tabulation output (eg from survey analysis), which makes it trivial to embed the results into DTP systems for reporting, although it ties you to using a PostScript printer.

Having started using a mainframe version, the PC port is a little slower than expected, even on a fast 386, mainly because of DOS's file handling, and the need for a heavy overlay structure. It takes mostly the full remains of your 640 KB, but it does run very happily under DESQView.

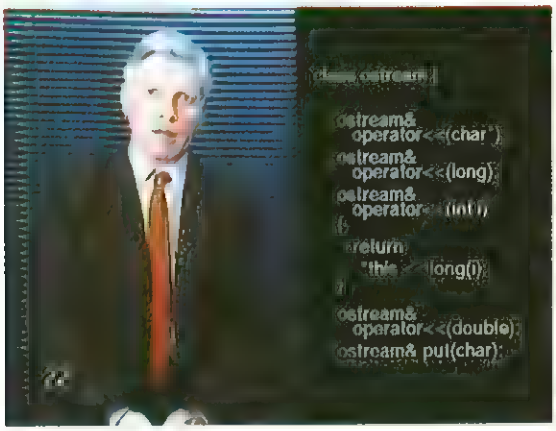
The documentation is a dog at first sight: a whacking 3-volume manual, each one an inch thick, but turns out to be clearly written

```
modify animals[if slaughter.date good, retain;
       keep pig.no ?.weight slaughter.date],
   out dead.animals$
```

*Figure 2 - PPL without the stats*

# VIDEO COURSE

# Learn C++




**NO RUSH!**
**NO TRAVEL!**
**NO HOTELS!**
**ONLY £299.95**

### Learn C++ Now!

The great exodus of programmers from C to C++ has begun! Since C++ builds on C, it's the easiest OOP language to learn. That's why it's called "the language of the 90's".

### Why the rush?

PRODUCTIVITY! Yes, C++ programmers can write programs in less time requiring less maintenance. Large projects become much easier to manage.

Unfortunately, learning C++ can be very costly. Classroom instruction is expensive even without the travel and hotel costs. Of course, not learning will cost you even more in the long run. Now there is an alternative!

### The top C++ video tutorial at the lowest possible price.

The C++ video tutorial from Zortech is the ultimate C++ training tool for work or home at only £299.95. It comes on six VHS video tapes containing 32 clear, extensive tutorials.

Used in conjunction with the concise workbook and tutorial disk, you will find everything you need for fast-track C++ tuition.

The course is generic (i.e. compiler and hardware independent) and is available with or without the award winning Zortech C++ Compiler for MS-DOS and OS/2.

### Start writing C++ code within a week.

As a C programmer, you will start producing C++ code within a week of concentrated use of this course. Alternatively, spend just an hour a day watching the video and working through the suggested exercises to learn C++ in only six weeks!

### The Leader in video tutorials.

If you don't already know C, you can join tens of thousands of programmers world-wide who have learnt C with the Zortech Complete C Video Course, described as:

"An excellent bargain ...
... I heartily recommend"
Gary Ray, PC WEEK

### Save your company thousands of £££'s.

One programmer can train for only £299.95, but you can train ten programmers for just:

| | |
|---|---|
| C++ Course | £299.95 |
| 9 Extra Workbooks | £179.55 |
| Total 10 students | £479.50 |

Yes! Only £47.95 each! (With all the FREE refresher courses you need!)

Now, Zortech with its new C++ Video Tutorial has refined the art of video tuition and presentation even further for the 1990's.

### Only £299.95 complete

- Six Videos with 32 lessons
- 256 page workbook
- Tutorial disk
- Compiler & hardware independent
- NTSC or PAL format
- Tax deductible

### Don't delay, order now!

Just mail the coupon or call the order hotline for same day shipment.

**USA:** Zortech Inc.,
1165 Massachusetts Avenue,
ARLINGTON, MA02174
Voice: 617-646-6703
Fax: 617-643-7969

**EUROPE:** Zortech Ltd.,
106-108 Powis Street, LONDON
SE18 6LU
Voice: 44- 1-316-7777
Fax: 44- 1-316-4138

**IN EUROPE CALL:**

## 44- 1-316-7777

**ORDER FORM**

Please rush me these items:

| Qty | Description | Price |
|---|---|---|
| ___ | C++ Video Tutorial | £299.95 |
| ___ | Extra C++ Workbooks | £. 19.95 |
| ___ | Zortech C++ Compiler | £129.95 |
| ___ | Developer's Edition | £299.95 |

Please add £5 + VAT postage in UK
All UK orders please add 15% VAT
International shipping charged at cost

Name _____
Company _____
Address _____
_____
_____
Phone _____
MC,VISA or CHQ_____ Expiry ____
Card No. _____ .EXE 9/91

CIRCLE NO. 160

# Ridgeway Press

13 Horseshoe Park
Pangbourne
Berkshire RG8 7JW

COMPUTER MANUAL

## PRINTING

*Specialist printers of*

**Technical Publications, Manuals, Instruction Books**

Ring binders/ slip cases plain or printed
Wire-o-binding, tabbing
Shrink wrapping, disk labels
New manuals or reprints. All sizes
Data conversion
Full artwork/origination service

## RAPID TELEQUOTE SERVICE

**RIDGEWAY PRESS, 0734 845331**
**FAX 0734 845186**

CIRCLE NO. 161

# Think BIG!

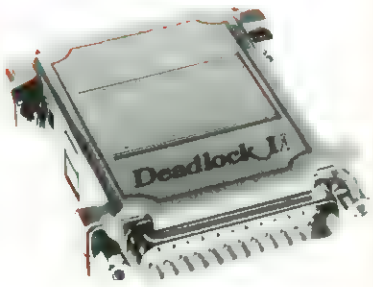

Forget 640K. Forget EMS. Forget the 16-bit squeeze. With LPA 386-PROLOG, you get a genuine 32-bit programming environment, which runs directly under MS-DOS and yet is capable of addressing up to 4G (4096M) of memory. All you need is a 386 or 486, and the talent to Think BIG!

Logic Programming Associates Ltd
Studio 4, Royal Victoria Patriotic Building
Trinity Road, London, SW18 3SX, England
Tel: 081 871 2016 - Fax: 081 874 0449

CIRCLE NO.162

## SPECIAL PROMOTION
# Deadlock-I

**ONLY £8.75**
Windows 3 supported
Quantity Order




- Software protection units that are programmable by the Software House.
- No need to keep a stock.
- Ten (10) days delivery guaranteed.
- The codes can be changed if desired by the software house.
- Supplied with **FREE** software.
- Evaluation kit available free for six weeks.
- Are you already using DONGLES? You can have us as a second source. (Call for details).

**Please address all enquiries to:**
**B.L. Computer Security, 101 Hendon Lane, Finchley, London N3 3SH, or call us on 081-343 0734 or Fax us on 081-346 2672**

CIRCLE NO. 163

# MAKE WORDPERFECT 5.1 MULTILINGUAL

## WorldFont for WordPerfect

- Uses standard WordPerfect 5.1
- WorldFont for WordPerfect includes: Arabic/Farsi*, Armenian, Europe set (includes Finnish, Danish, Swedish, Turkish, German, French, Spanish and Norwegian), Greek modern and ancient, Russian 1 & 2, mathematics 1 & 2, multinational 1 & 2, Polish, Hebrew*, Japanese Katakana/Hiragana, Thai, Typographic symbols
- Additional fonts available: Hungarian, Portugese, Greek Helvet type face, Cyrillic, Czech, special Japanese Katakana and Sanskrit. All languages include 8, 10, 12 and 14 point fonts. Languages with (*) also have 20 point fonts.
- Special Hebrew module (12 Type fonts available)
- Includes high quality bit-mapped laser fonts
- Includes .ALL files for most laser printers
- Does not modify WordPerfect 5.1 - so you have access to all WordPerfect commands
- Includes pop-up on-screen template indicating location of foreign language characters on the keyboard
- Single toggle between languages
- View and print multiple languages on the editing screen simultaneously within one document
- Right-to-left entry for Hebrew and Arabic
- Compatible with Bitstream and other bit-mapped laser fonts
- Not a TSR: it does not tie up RAM
- Optional WorldFont Plus package includes LaserTwin, a utility that prints laser fonts on most dot-matrix or InkJet printers
- Language Keytop Overlays are also available

**Only £149**

**£199 for WorldFont Plus for WordPerfect 5.1 (includes LaserTwin)**

**THE TEXT FORMATTING COMPANY**
Suffield Works, 1 Suffield Road, London N15 5JX Tel: 081 802 4470

CIRCLE NO.119

with plenty of examples. I've done up my own crib on two sides of A4 for day-to-day use.

## Triangulation

The Triangle Problem turned out to be trivial: the `BUILD` command in its default format prompts the user for the named variables, case after case, until a dollar sign is entered. To save scrolling, multiple records can be entered on a line, using a slash to delimit them. The `PROCESS` command can then be used to do the testing of the sides.

Non-triangularity is tested with three `OR`'d conditions. Equilateral and isosceles triangles are detected by using the `ALL` and `ANY` operators to test successive argument variables within parentheses for equality with the right-hand side of the condition.

I think PPL's compactness of expression makes it one of the shorter Triangle Problem solutions: it was certainly one of the easiest to write.

P-Stat is not going to be for everyone. A lot of people are put off by command line driving, having been brought up on menuing word processors and spreadsheets; and not everyone wants to give over 5MB of hard disk for data manipulation (odd, though, they'll happily give over 5MB to a word processor in which they only use 10% of the facilities!).

But if you want robust and reliable calculations on bulk data, particularly if it is a repetitive production task (there is very good batch support including the use of command files), with presentation-quality listings or tabulations and *ad hoc* manipulation facilities, then P-Stat is well worth looking at.

EXE

---

*Peter Flynn is currently manager of the research and academic computing development service at University College, Cork. He is into early music, reading, surfing, typography and cyberspace. He can be reached by email as `pflynn` on BIX and CIX, through the wide-area networks as `pflynn@iruccvax.ucc.ie`, and as `silmaril` on Relay and IRC. He was Chairman of the European P-Stat Users Group for several years, but no-one held it against him.*

*P-Stat comes from P-Stat Inc, PO Box AH, Princeton Nj 08542, USA. The UK agent is Timberlake-Clark Ltd, 40B Royal Hill, Greenwich, London SE10.*

```
build triangle, vars a b c;
 1 1 4 / 1 4 1 / 4 1 1 / 3 3 3 / 2 3 3 / 2 3 4 $

process triangle[if a+b<c or a+c<b or b+c<a,
                 put a b c ' is not a triangle', goto done;
        if all(b c)=a, put a b c ' is equilateral', goto done;
        if any(b c)=a or b=c, put a b c ' is isosceles',
           goto done;
        put a b c ' is scalene';
        done:]$
```

*Figure 3 - PPL used on the Triangle Problem*

# The Case of the Senile Rodent

*Does your mouse behave strangely with certain packages? Dave Midgley discovered that some elderly mice are not quite all there, and provides a restorative.*

Anyone who is brave enough to use the new Borland C++ package on their Amstrad 1640 will have discovered, as I did, that as well as the size and speed constraints there is also a teensy problem with the mouse. Everything seems to work fine until you try and shell to DOS or to exit the program. Shelling to DOS causes the cursor to disappear forever, and exiting, although apparently harmless, leaves the interrupts in a state that causes the machine to crash completely the next time you move the mouse.

This is all due to the fact that many newer packages, notably Borland IDEs, require functions which are not implemented in certain older mouse drivers, notably Amstrad's. I mention Amstrad in particular because the driver for the 1640 has not been brought up to date even in the very latest version.

## The solution

The problem can, however, be solved by the simple expedient of writing a small TSR which hangs on interrupt 33h and provides the missing functionality to the mouse interrupt handler. The functions which are missing (you can check these in your mouse documentation) are 14h - swap user defined subroutine mask, 15h - get storage size, 16h - save driver and 17h - restore driver.

The first of these, swap user defined subroutine mask, relates to function 0Ch which sets up a user subroutine to be called whenever certain mouse events occur. Function 14h swaps the current user event mask and address with new ones provided in CX and ES:DX, and returns the old mask and address, likewise in CX and ES:DX. The other three functions allow the complete state of the mouse driver to be saved and subsequently restored.

To do all this the mouse state must be stored in a data structure containing the user-defined call mask, the user-defined call far address, the cursor state (on or off) and the maximum and minimum values for the cursor X and Y positions. Function 15h then returns the size of the structure, function 16h copies the whole structure into a user buffer addressed by EX:DX, and function 17h restores the mouse state from a user buffer, likewise addressed by ES:DX.

Restoring the mouse state in function 17h is not, of course, just a matter of copying the data into the structure. The mouse must be reset and the various parameters reset by calls to the original mouse driver. (Note that when the original interrupt address is called the flags

```
VERSIONSTR      equ  "1.01x"
code SEGMENT    public
     ASSUME     cs:code, ds:code, ss:code
MOUSEINT  equ  33h  ; mouse interrupt number
public    mousepat
MOUSEBUFFER     STRUC
callMask        dw   0
eventSeg        dw   0
eventOff        dw   0
cursorState     dw   0    ; 1 is visible
cursorMinX      dw   0
cursorMaxX      dw   639
cursorMinY      dw   0
cursorMaxY      dw   199
ENDS
     ORG  5Ch  ; use FCB for non-preset data
; old interrupt address
oldInt     LABEL     dword
oldOff     dw   ?
oldSeg     dw   ?
cxTmp      dw   ?    ; temps used by swapUser
esTmp      dw   ?
dxTmp      dw   ?
     ORG  100H ; code entry for .COM
mousepat:
     jmp  begin     ; jump to transient
     ; preset data :
; mouse data buffer
mouseBuff MOUSEBUFFER    ?
cursorFlag      dw   -1   ; counts hides/shows
; table of computed jumps
JumpTable LABEL     near
     dw   resetMouse        ; code 00
     dw   showCursor        ; code 01
     dw   hideCursor        ; code 02
     dw   oldRoutine        ; code 03
     dw   oldRoutine        ; code 04
     dw   oldRoutine        ; code 05
     dw   oldRoutine        ; code 06
     dw   setXMinMax        ; code 07
     dw   setYMinMax        ; code 08
     dw   oldRoutine        ; code 09
     dw   oldRoutine        ; code 0A
     dw   oldRoutine        ; code 0B
     dw   setUser           ; code 0C
     dw   oldRoutine        ; code 0D
     dw   oldRoutine        ; code 0E
     dw   OldRoutine        ; code 0F
     dw   OldRoutine        ; code 10
     dw   oldRoutine        ; code 11
     dw   oldRoutine        ; code 12
     dw   oldRoutine        ; code 13
     dw   swapUser          ; code 14
     dw   getSize           ; code 15
     dw   saveDriver        ; code 16
     dw   rstDriver         ; code 17
;  The new interrupt 33h - ISR entry point.
newInt    PROC
     jmp  overID
     db   "MOUSEPAT"     ; ID footprint
overID:
     push bx
     cmp  ax, 17H   ; highest in table is 17h
     jnle oldRoutine
     push ax
     shl  ax, 1
     lea  bx, jumpTable  ; get address in bx
     add  bx, ax
     pop  ax
     jmp  WORD PTR cs:bx
newint    ENDP
; WARNING, bx left on stack -
; must be popped by every function
; The new Interrupt 033h routine -
;         extension and additional functions
oldRoutine     PROC
     pop  bx   ; lose computed jump address
     jmp  cs:oldInt
oldRoutine     ENDP
;    Extension to function 00h (Reset Mouse)
resetMouse     PROC
     pop  bx   ; lose computed jump address
     mov  cs:mouseBuff.callMask, 0
     mov  cs:mouseBuff.eventSeg, 0
     mov  cs:mouseBuff.eventOff, 0
     mov  cs:mouseBuff.cursorState, 0
     mov  cs:mouseBuff.cursorMinX, 0
     mov  cs:mouseBuff.cursorMaxX, 639
     mov  cs:mouseBuff.cursorMinY, 0
     mov  cs:mouseBuff.cursorMaxY, 199
     mov  cs:cursorFlag, -1
     jmp  cs:oldInt
resetMouse     ENDP
; Extension to function 01h (Show Cursor)
showCursor     PROC
     pop  bx   ; lose computed jump address
     inc  cs:cursorFlag
     jnz  endShow
; ...set cursor visible...
     mov  cs:mouseBuff.cursorState, 1
endShow:
     jmp  cs:oldInt  ; execute old routine
showCursor     ENDP
; Extension to function 02h (Hide Cursor)
hideCursor     PROC
     pop  bx   ; lose computed jump address
     dec  cs:cursorFlag
     jnc  endHide
; set cursor invisible
     mov  cs:mouseBuff.cursorState, 0
endHide:
     jmp  cs:oldInt ; execute old routine
hideCursor     ENDP
; Extension to function 0Ch (Set User Event)
setUser   PROC
     pop  bx   ; lose computed jump address
; store user event data
     mov  cs:mouseBuff.callMask, cx
     mov  cs:mouseBuff.eventSeg, es
     mov  cs:mouseBuff.eventOff, dx
     jmp  cs:oldInt  ; execute old routine
setUser   ENDP
; Extension to function 07
; (Set Minimum and Maximum X cursor position)
setXMinMax     PROC
     pop  bx   ; lose computed jump address
     mov  cs:mouseBuff.cursorMinX, cx
     mov  cs:mouseBuff.cursorMaxX, dx
```

*Figure 1 - MOUSEPAT.ASM*

must be pushed first with `pushf`. This is because an interrupt pushes both flags and return address, and the `iret` at the end of the service routine pops them.)

Unfortunately that is not the end. In order to maintain the information about the mouse state, several of the other functions must be trapped as well. Function 0 (reset) must also reset the state information; functions 1 and 2 (show and hide cursor) must update the cursor state, bearing in mind that multiple shows require multiple hides and vice versa; function 0Ch, like the new function 14h, must save the user-defined call mask and address; and finally functions 07h and 08h must save the maximum and minimum values of X and Y. In each of these cases the routine finishes with a jump to the original mouse driver.

Figure 1 shows the assembler code implementation of the patch. This is not the place to go into detail on how to write TSRs, so I shall assume you know the principles and just give a brief overview of the program.

The mouse state information is stored in a structure called `MOUSEBUFFER`, instantiated as `mouseBuff`. Storage for the old interrupt vector and a swap area for function 14h come next. I always put uninitialised data in the FCB area between 5Ch and FFh. (This is quite legal and saves space in larger TSRs). Because it is a .COM file execution must start at 100h. A jump to the start of the transient code is put at this address followed by any initialised data - the mouse state buffer and the count which is used to match cursor shows and cursor hides.

I used an address table and a computed jump to decode the interrupt opcode in `al`. The addresses are in `jumpTable`, and `newInt` is the new interrupt routine address. Note the 'footprint' placed at the beginning of the interrupt routine, which is used by the transient code to check if the extension is, in fact, already loaded. This is followed by the six extension functions and the four new functions plus a straight-through function for untrapped calls.

The transient code displays its name and version, checks that there is a mouse driver loaded, but that the extension has not been loaded, sets up the new interrupt vector and terminates, leaving the new routines resident.

The program is designed as a .COM (easier for TSRs), so the linker will give a message 'no stack segment' which you can ignore. After linking, convert to .COM with EXE2BIN.

EXE

*Dave Midgley is a free-lance software engineer, just trying to earn an honest buck in these desperate times.*

```
     jmp  cs:oldInt
setXMinMax     ENDP
; Extension to function 08
; (Set Minimum and Maximum Y cursor position)
setYMinMax     PROC
     pop  bx   ; lose computed jump address
     mov  cs:mouseBuff.cursorMinY, cx
     mov  cs:mouseBuff.cursorMaxY, dx
     jmp  cs:oldInt
setYMinMax     ENDP
; Additional function 14h
; (Swap User Defined Subroutine Mask)
; On entry registers contain user event data
; as for function 0Ch.
; On exit registers contain old user event
; data.
swapUser  PROC
     pop  bx   ; lose computed jump address
; put current user event into temporary
store
     mov  ax, cs:mouseBuff.callMask
     mov  cs:cxTmp, ax
     mov  ax, cs:mouseBuff.eventSeg
     mov  cs:esTmp, ax
     mov  ax, cs:mouseBuff.eventOff
     mov  cs:dxTmp, ax
; save new user event
     mov  cs:mouseBuff.callMask, cx
     mov  cs:mouseBuff.eventSeg, es
     mov  cs:mouseBuff.eventOff, dx
; call old set event routine
     mov  ax, 0Ch
     pushf     ; (to simulate interrupt)
     call cs:oldInt
     mov  ax, 14h       ; replace call code
; return old user event
     mov  cx, cs:cxTmp
     mov  es, cs:esTmp
     mov  dx, cs:dxTmp
     iret
swapUser  ENDP
; Additional function 15h (Get Storage Size)
; Buffer size returned in BX
getSize   PROC
     pop  bx   ; lose computed jump address
; return size of buffer
     mov  bx, SIZE MOUSEBUFFER
     iret
getSize   ENDP
; Additional function 16h (Save Driver)
; Mouse data is moved to location ES:DX
saveDriver     PROC
     pop  bx   ; lose computed jump address
     push cx
     push si
     push di
     push ds
     mov  cx, SIZE MOUSEBUFFER
     lea  si, cs:mouseBuff
     mov  di, dx
     mov  ax, cs
     mov  ds, ax
     cld
     rep  movsb     ; copy all data to ES:DX
     pop  ds
     pop  di
     pop  si
     pop  cx
     mov  ax, 16h   ; replace call code
     iret
saveDriver     ENDP
; Additional function 17h (Restore Driver)
; Restores mouse functionality. Mouse data
; is moved from location ES:DX
rstDriver PROC
     pop  bx   ; lose computed jump address
     push cx
     push dx
     push ds
     push es
     push si   ; restore mouse data
     push di
     mov  cx, SIZE MOUSEBUFFER
     mov  si, dx
     lea  di, cs:mouseBuff
     mov  ax, es
     mov  ds, ax
     mov  ax, cs
     mov  es, ax
     cld
     rep  movsb
     pop  di
     pop  si
     mov  ax, 0     ; reset mouse
     pushf     ; simulate interrupt
     call cs:oldInt
     mov  cs:cursorFlag, -1
     mov  ax, 4     ; reset position
     mov  cx, 0
     mov  dx, 0
     pushf     ; (simulate interrupt)
     call cs:oldInt
     mov  ax, 0Ch   ; restore user event
     mov  cx, cs:mouseBuff.callMask
     mov  es, cs:mouseBuff.eventSeg
     mov  dx, cs:mouseBuff.eventOff
     pushf     ; (simulate interrupt)
     call cs:oldInt
     mov  ax, 07    ; restore min & max X & Y
     mov     cx, cs:mouseBuff.cursorMinX
     mov  dx, cs:mouseBuff.cursorMaxX
     pushf     ; (simulate interrupt)
     call cs:oldInt
     mov  ax, 08
     mov     cx, cs:mouseBuff.cursorMinY
     mov  dx, cs:mouseBuff.cursorMaxY
     pushf     ; (simulate interrupt)
     call cs:oldInt
; if cursor visible ...
     cmp  cs:mouseBuff.cursorState, 0
     jz   rstEnd
; ... make it visible
     mov  ax, 1
     pushf     ; (simulate interrupt)
     call cs:oldInt
rstEnd:
     pop  es
     pop  ds
     pop  dx
     pop  cx
; restore opcode for tidiness
     mov  ax, 17h
     iret
rstDriver ENDP
;    The transient bit.
transient:
signOnTxt db   0Dh, 0Ah
          db   "Mouse driver extension,"
          db   " Version ", VERSIONSTR
          db   0Dh, 0Ah, 0Dh, 0Ah, "$"
IDTxt     db   "MOUSEPAT"   ; for comparison
noMouseTxt db  "Error - original mouse driver"
           db  " not loaded"
           db  0Dh, 0Ah, "$"
duplTxt   db   "Error - Mouse extension "
          db   "already loaded",
          db   0Dh, 0Ah, "$"
OKTxt     db   "Mouse extension loaded"
          db   0Dh, 0Ah, "$"
begin     PROC ; execution start
     lea  dx, signOnTxt
     mov  ah, 09h   ; write string to console
     int  21h
     mov  al, MOUSEINT
     mov  ah, 035h  ; get interrupt vector
     int  021h ; ... and check for ...
     mov  ax, es    ; ... vector = 0 or ...
     or   ax, bx
     jz   noMouse
     mov  al, es:bx
     cmp  al, 0CFh  ; ISR is just RETI
     jz   noMouse   ; ie. no mouse loaded.
     mov  di, bx    ; look for footprint
     add  di, 3     ; skip 'jmp' instruction
     lea  si, cs:IDTxt
     mov  cx, 8
     repz cmpsb
     jz   dupl ; patch already loaded
     mov  cs:oldSeg, es  ; replace interrupt
     mov  cs:oldOff, bx
     mov  al, MOUSEINT
     lea  dx, newInt
     mov  ah, 025h  ; set new vector
     int  021h
     lea  dx, OKTxt
     mov  ah, 09h   ; write string to console
     int  21h
; Calculate no of paragraphs in resident bit
     mov  dx, OFFSET transient
     add  dx, 15h
     mov  cl, 4
     shr  dx, cl
     mov  al, 0     ; set result = OK
     mov  ah, 031h
     int  021h ; terminate and stay resident
noMouse:
     lea  dx, noMouseTxt
     mov  ah, 09h   ; write string to console
     int  21h
     mov  al, 1     ; set error condition
     mov  ah, 04Ch
     int  021h ; terminate
dupl:
     lea  dx, duplTxt
     mov  ah, 09h   ; write string to console
     int  21h
     mov  al, 1     ; set error condition
     mov  ah, 04Ch
     int  021h ; terminate
begin     ENDP
code ENDS
END  mousepat
```

*Figure 1 - MOUSEPAT.ASM (Continued)*

# UNIX tools on DOS

*Peter Collinson has been experimenting with a toolkit which makes DOS seem more friendly to the UNIX user - and also offers something for the native DOSser.*

The MKS toolkit is a collection of DOS programs that emulate equivalent UNIX tools. The programs are managed by a clone of the UNIX Korn shell. I wasn't very happy about the Coherent UNIX look-alike for DOS machines, largely because its tools were really based on UNIX Version 6. Frankly, things have moved on from there. The MKS toolkit doesn't suffer from these problems: the tools have tracked UNIX closely. The result is a set of programs running on DOS that provide UNIX look, feel and functionality.

Authors of UNIX look-alike tools often fail to avoid the temptation to 'improve' them. MKS wins because the company has rejected improvement in favour of compatibility. Sometimes the tools have been altered to cope better with the DOS environment. The changes tend to be enhancements rather than alterations in original functionality.

## Tools

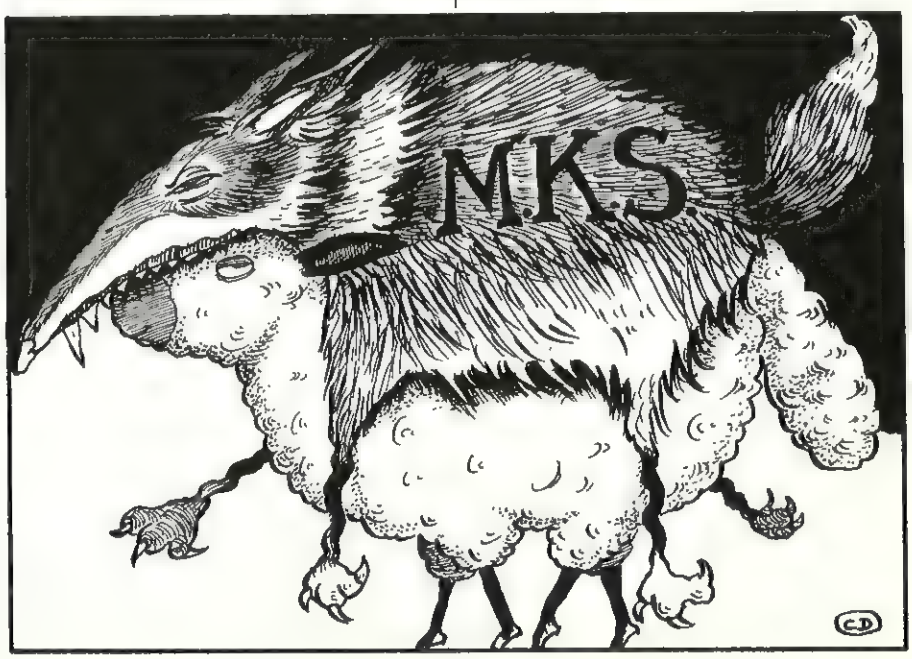
When you set out to build a set of UNIX tools, you quickly get past the first hurdle of writing the basic file manipulation set. Things like copy a file (`cp`), move a file (`mv`), delete a file (`rm`) and list a directory (`ls`) are reasonably trivial to code. I had these for DOS from public sources well before I acquired my copy of MKS. They are easy to write, but it's harder to make them work exactly like their UNIX counterparts. Copying a file to a new named file is easy. Copying a file or a list of files to a directory is more difficult; especially when you consider that DOS treats the top directory on any file system as something special.

After you write the basic set, then there are several tools that really ought to be on a UNIX system. These are technically harder to write but well within the reach of most programmers. For example, there are tools for comparing files, like `diff` or `cmp`. There are programs for looking for strings in files: the `grep` family. There are loads of tools that are designed to process text files: to display them on the screen; to look at their start or their end; to count characters, words or lines; to sort them; or to do simple formatting. Many of these programs are not used every day but are part of the UNIX tool-set, to be used when occasion demands.

Finally, there are a set of tools that are just hard to write because the task to be done is complex; or perhaps the original was large or idiosyncratic. MKS provides several tools in this category and they are well done too. First, MKS provides the UNIX supertools: `awk` and `sed`. These are an invaluable help to the shell programmer. They allow numeric and text data contained in text files to be processed without resorting to a compiled programming language like C.

MKS supplies the two main forms of file archiver, `tar` and `cpio`. These were originally developed to write tapes but are probably most used these days to create a single file containing several other files. I tend to store sources on my machine in compressed `tar` format, using the `compress` program to minimise disk occupancy. The MKS versions of both `tar` and `cpio\f` have `compress` and `uncompress` built in. This reflects the inability of DOS to support pipes *properly*. Oh, MKS gives you working versions of `compress` and `uncompress` too.

MKS supports the two standard UNIX editors: `ed` and `vi`. The `ed` command is the original UNIX line editor. It's simple and fast. The `vi` command is a full screen editor. If you are used to word processors then you may find `vi` hard to learn. There is a large-ish learning curve: you simply have to train your body to remember what keystrokes are used to do what. If you are a UNIX user struggling to use a DOS machine, and wondering whether to learn the arcane mysteries of `EDLIN`, then you will probably feel relieved to find that the editor that runs on every UNIX system is now available on DOS.



# Spend two free hours with Sycero dB and save days of development time

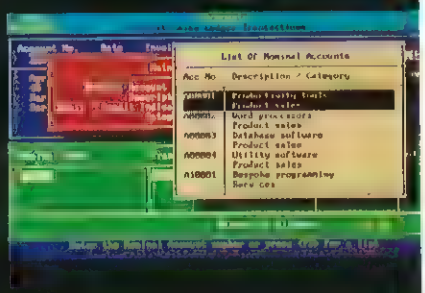

Imagine being able to develop applications quickly, simply and reliably — with the minimum of debugging. And in a fraction of the time. But without losing any of the flexibility.

Now you can. With Sycero dB, a complete and powerful program generator that lets you develop applications in a fraction of the time it takes to do manually — by automating most of your programming. What's more, Sycero dB actually lets you add code and subroutines *before* generation. So you have the same flexibility that you value with manual coding.

Clipper is the applications development standard for PC database management systems. Its open architecture and superior networking capabilities let you create really sophisticated applications.


1


2




3

Sycero makes use of Clipper's advanced facilities, and goes a stage further. Sycero dB NET actually builds in the automatic file and record locking required by network systems.

Together, Sycero and Clipper form the perfect development environment. But don't just take our word for it. Seeing is believing, and time is money. So spend just two hours with System C at one of our free Sycero dB seminars.

By building a suite of programs from scratch, we'll demonstrate how you can use Sycero to utilise the same development tools for both simple and complex systems; ensure complete programming standards; take the work out of networking, *and* save days of development time.

We know you'll be impressed with what you can do with Sycero in just a couple of hours. So think how productive you could be if you were using it all the time.

To find out more, simply complete and return the coupon, or contact System C on 0622 691616. It's your entry point to extra power and productivity.

1. *Create pull-down, bounce-bar and Novell-style menus in minutes.*
2. *Use memo fields to give you on-screen word processing in a window.*
3. *Overlay a browse window with a function key in just five lines of code.*
4. *A complete and powerful report generator for forms, lists and more complex invoices/statements.*
5. *Transactions or multiple records per screen produced easily.*

4

5

Nantucket®

*All trademarks are acknowledged.*

YES. I want to know how to create more powerful and sophisticated applications.
Send me more information about ❑ Sycero dB ❑ your next Seminar ❑ the System C product range

Name ..... Position .....
Company .....
Address .....
.....
Post Code ..... Tel No .....

I am currently using the following products:
.....
.....

EXE 9/91

Please return to: System C Ltd., 60/61 High Street, Maidstone, Kent ME14 1SR. Tel: 0622 691616 Fax: 0622 691241

CIRCLE NO. 166

## Shells

The MKS tools are all normal DOS programs that can be picked up and run. If you wish, you can continue to use COMMAND.COM. The programs will just work providing you with functionality that is missing on DOS systems. All this is relevant if you find the idea of making one system look like another is anathema to you.

However, I opt for compatibility with UNIX and run MKS's Korn shell as my command interpreter. It is not only a command interpreter, it is also a complex programming language allowing many tasks to be done by interpreting text files, often called scripts.

The proliferation of UNIX shells can be confusing. In the past there have been two main variants: the Bourne shell (sh) and the C-shell (csh). The Bourne shell was first supplied with UNIX Version 7, the great-great-grandparent of all UNIX systems today. As a consequence, sh is available on all UNIX systems.

Berkeley generated csh and this is available on all systems derived from BSD releases. The shell supports Berkeley system features like job control and also contains a more friendly user interface. For example, the shell remembers the commands that the user has typed and this history can be accessed to create new commands. Recently, csh has acquired another goody: filename completion. If you are typing a file name on a command line, then you only need to type the first few characters and hit escape. The shell looks in the directory to match the filename and will supply you with the remaining characters if the stem is unique. If the stem that you have typed is ambiguous the shell beeps to invite you to type more. For several years, I typed into my UNIX machines using csh but wrote shell scripts using sh.

Then came the Korn shell. It is based on sh syntax and is supposed to be interchangeable with the Bourne shell as far as scripts are concerned. However, it *is* another language and has many extra control constructs that solve some of the problems with the Bourne shell. For instance, there is a built-in command called select that takes a set of strings and prints a menu for the user to choose from.

The shell comes along with filename completion and history editing. You can use vi or emacs style editor commands to retrieve a previously typed command. You then change it to a new edited version before hitting return to run the commands in the line. This is much more natural than the way that csh did things - it's closer to the line editing offered by VMS or the COMMAND.COM program. Although, I might add, it's a lot more powerful than either of those.

***vi has a large-ish learning curve; you have to train your body to remember the keystrokes***

## The MKS shell

The MKS Korn shell is indistinguishable from its UNIX counterpart. When running it on your DOS machine, you can begin to make believe that you are using UNIX. UNIX mimicry is further enhanced by the switch program. This changes the usual DOS option character from forward slash '/' to a minus '-'. DOS then understands that forward slashes in filenames are to be thought of as separators, just like UNIX.

The shell can be started interactively from COMMAND.COM just like a normal command. It then takes control of the machine acting as your command interpreter. I do this on the 386 machine that I use, because it mostly acts as a straight DOS engine running standard applications.

Things become more interesting when you integrate a call to the Korn Shell into the normal bootstrap of the system. First, you can add a line into AUTOEXEC.BAT calling the shell. An option to the shell tells it that when a new command is invoked it should free much of the memory it needs to run in. It hangs onto a small amount so a reloader can start the shell again after the command has finished. This is just like COMMAND.COM.

This solution does retain a portion of COMMAND.COM in memory. If this is a pain, you can replace COMMAND.COM completely by starting the shell from CONFIG.SYS. I cannot use this method on my laptop because I have found that it depends on drive C: containing the MKS start-up configuration files. I cannot write the files there because drive C: is a ROM. I use the previous setup, running the shell from AUTOEXEC.BAT.

Finally, you can go the whole hog and login to your DOS machine just like UNIX. You run a program called init from the bootstrap sequence. This will run any special start-up programs and load any TSRs. Finally, it runs a login program that asks you for a password. This last option is great if several different people use the 'personal' computer, because each user can have a different environment after they have logged in. The login program simply acts as a key into the system and does not provide file access control like UNIX.

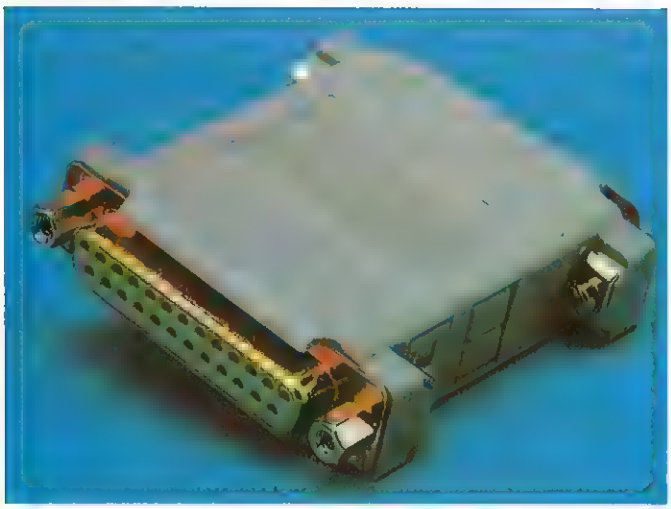
## Documentation

The MKS toolkit is accompanied by some impressive documentation. First, there is a small booklet that explains how the toolkit is to be installed. This gives details of the four installation methods that I touched on above. This booklet is also the place that MKS prints release notes and a copy of the README.1ST file. Finally, there are the answers to a number of frequently asked questions.

The second document is a *User's Guide*. It consists of 250 pages. The book is spiral bound so that it lays down flat on a surface leaving hands nicely free for typing. It consists of a number of introductory articles intended as tutorial material. The first, Overview, gives a brief description of the basic tools: how they work and inter-operate. There are then lengthy introductions to awk, the Korn shell and vi. The book ends with a useful glossary and a comprehensive index.

The third book is a *Reference Manual*. The emphasis here is documenting what the commands really do rather than how they are used. The book is a little over 450 pages and is also spiral bound. It is printed very much in the style of UNIX manual pages where each command has a 'page' to itself. Each 'page' can be several physical sides of paper. The pages are sorted by command name so finding the entry for a particular command is easy.

A manual page has several well defined sections that always appear in the same order. The first section is *NAME*, and contains the name of the command and a one line description of what the command does. The second section is *SYNOPSIS* showing what you type to use the command. This is where you look if you are just checking on what options a command has. Next there is a lengthy section called *DESCRIPTION* containing a narrative on what the command does and how the options change its action. For a complex command like awk the description covers several pages and has several subsections.



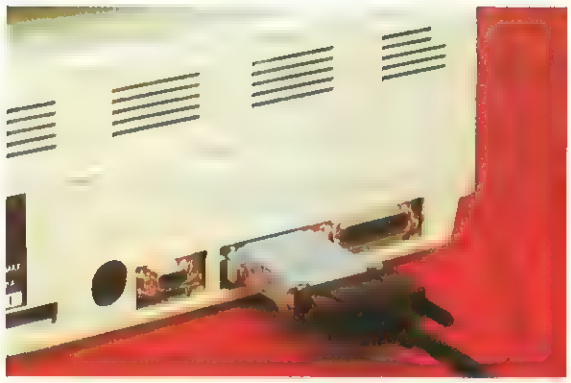
How many users of your software paid for it???

SOFTLoK International Limited was established in 1987 with the introduction of our SOFTLoK and SOFTLoK PLUS devices to combat the ever increasing problem of software piracy. Our SOFTLoK range of software protection devices are used by hundreds of software developers from small consultancies to large multinationals. To cope with the ever changing needs of our customers we have developed SOFTLoK II which combines the programmable features of SOFTLoK PLUS with a low unit cost similar to our original SOFTLoK product.

# SOFTLoK II™

# The Next Generation

SOFTLoK II units are programmable devices containing read/write memory protected by a password. Both the memory and the password can be changed at any time using our routines in your application software. Easy to use menu-driven software is provided to allow small or large batches of SOFTLoK II units to be programmed with their initial data & passwords ready to be sent out with the protected software product.



*As SOFTLoK II units plug into the parallel printer port they can be installed or removed in seconds.*

Price: 1-19 £16.90, 20-49 £15.40, 50+ £13.70

Evaluation kit £20 (SOFTLoK II, manual & software)
All prices excl. VAT and delivery

- For IBM PC, PS/2 and compatibles
- Uses parallel printer port
- Totally transparent to printer
- Secure data & password can be changed from your application software
- Cascadeable
- 240 bytes of secure read/write memory
- 8 byte (64 bit) password
- No programming adaptors required
- Easy to use SOFTLoK II setup software
- Routines ready to link with various compilers
- Easy to follow manual

I-MEX House, 40 Princess Street,
Manchester, M1 6DE, England,
**Tel:** 061 228 7379 **Fax:** 061 236 6890

Menu-IX | System-IX | DBMS-IX | Sysix | AI | Spell | Merge | Lock | GURU | Crypt | Tawk | Tgrep

Cut | More | Term Ctl | Where | Sort | Grep | Idle | CB Trees | Evaluate | Awk | Banner | C Source | Fgrep | Help | Head | Tail

Xenix | Xenix Net | Unix | Interactive | System V | TCP/IP | DOS | Token-Ring | OS/2 | Ethernet | Novell Lan | Mac Net | Aix | VP-IX | VMS | QMS


Xvale Ltd., Store 'D'
55, Bedford Court Mansions,
Bedford Avenue, London.
WC1B 3AD U.K.

**TEL: 071-636-8210 / 071-631-4818**
**FAX: 071-255-1038 24 Hours**


**Barclaycard Access**

# UNIX

**Now! you can get into UNIX and VPIX Multi-user DOS) at affordable prices, with a leading specialist of the UK.**
**System V R 4: 8 console usr; BASE system, FACE, NFS,TCP/IP, RFS,DFS RPC,XDR,X11,Xt,Xview,Motif,(compat:BSD,XENIX),(Standards:SVID, POSIX,FIPS,XPG3),SDS,SCDE,DLL,ELF,FMLI,386/486/EISAsupported.**

## Interactive Unix SysV, Rel 3.2 Architect Series: (Ver 2.21)

**S.V.R.4. CALL !**

**Workstation Developer:** 1-2 usr £1295.00 U/L usr £1995.00
Operating System, Vpix (DOS under UNIX),
Software developement system plus New 'C', CO-edit, Codewatch,
Ten-Plus, Looking glass, XII windows runtime and development system,
NFS/TCP/IP, and complete documentation.

**Workstation Platform:** 1-2 usr £855.00 U/L usr £1495.00
As workstation developer, excluding software development.

**Network Developer:** 1-2 usr £1155.00 U/L usr £1815.00
As workstation developer, excluding XII Windows.

**Network Platform:** 1-2 usr £695.00 U/L usr £1350.00
As Network Developer, excluding development items.

**Application Developer:** 1-2 usr £1095.00 U/L usr £1750.00
As Network Developer excluding network items.

**Application Platform:** 1-2 usr £495.00 U/L usr £1095.00
As Application Developer, excluding development items.

| UNIX ITEMS | |
|---|---|
| Motif Dev | £345.00 |
| PC-DOS bridge | £175.00 |
| Norton for unix | £195.00 |
| Easy windows | £595.00 |
| Ported Netware | £3995.00 |
| SMB/ix | £895.00 |
| Informix/Oracle | Call |
| Lotus 123 | £495.00 |
| LPI Cobol | £1095.00 |
| LPI New 'C' | £595.00 |
| LPI Fortran | £ 895.00 |
| LPI Pascal | £895.00 |
| LPI Basic | £595.00 |
| LPI /PL/1 | £1795.00 |
| LPI RPGII Dev | £1795.00 |
| LPI RPGII Util | £895.00 |
| MENUIX | £49.95 |
| SYSIX -pro | £145.00 |
| LPI (unix) C++ | CALL |
| XSIGHT | CALL |

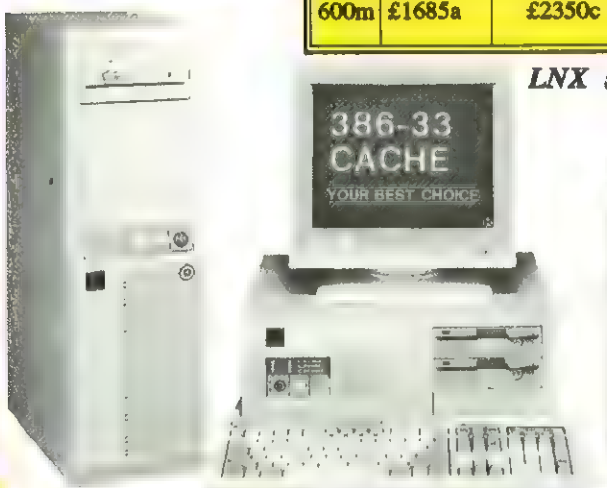
| SCO & Others | |
|---|---|
| Xenix Items | Call |
| Unix o/sys | £525.00 |
| Unix s/dev | £613.00 |
| Unix vpix | £321.00 |
| Unix tcp/ip | £227.00 |
| Unix nfs | £340.00 |
| Open dsktp | £795.00 |
| O/dtp s/dev | £1095.00 |
| O/dtp servr | £1095.00 |
| Guru reprter | £49.95 |
| Syseditor | £49.95 |
| Csheil | £49.95 |
| Menuix | £49.95 |
| Sysix Pro | £145.00 |
| SystemIX | £295.00 |
| PC-Interface | £170.00 |
| Pc-int/wndw | £225.00 |
| TCP/IP dos | £185.00 |
| PC-Xview | £245.00 |
| Termini ctl | £695.00 |
| PC-Xsight | £395.00 |

# LNX

**Base 1 level:** Mini-tower, Mono screen & adapter with parallel port, Dual hd/fd ctrlr with 2S/1P/1G port, 102 keyboard, 1 Mb ram, 1 * floppy drive (5-1/4 or 3-1/2), Mouse.
**Base 2 level:** as base 1 + SVGA colour screen & 512k svga card, 4 Mb ram on m'board.
**Base 3 level:** as base 2 + 1 Mb ram on SVGA card, 8 Mb ram on m'board.
**Base 4 level:** as base 3 + 16 Mb ram on m'board, Full tower case. (e & oe)

**FREE! SOFTWARE BUNDLES**
a: Menuix, Unix C-shell & tools on dos
b: Unix v2.02, Vpix, Menuix.
c: Unix v2.02 Application plat,Menuix
d: Unix v2.02 Network plat,Menuix
e: Unix v2.02 Worksn plat,Menuix
f: Unix v2.02 Workstation dev,Menuix
g: Unix v2.21 Workstation dev,Menuix
h: Unix ** System V rel 4 **,Menuix

| | 386-20 base 1 | 386-25 base 2 | 386-33 base 2 | 486sx33 base 3 | 486-33 base 3 | 486-50 base 4 | Unix Note book |
|---|---|---|---|---|---|---|---|
| 80mb | £1040a | £1610b | £1795b | £1945c | £2795e | £3695e | |
| 120m | £1110a | £1775b | £1860c | £2015d | £2935e | £3835e | £2495c |
| 200m | £1185a | £1850b | £2025c | £2175d | £3050e | £3950f | £g call |
| 600m | £1685a | £2350c | £2625d | £2775e | £3850f | £4850f | £h call |




*LNX systems*

*Unix Notebook - 20*

Option A
150 MB TAPE
£795

Option B
8 Port i/o with 2 * Wyse Terminal
£1395

Option C
1.2 Gigbt h/d & ctrl
£2495

Compaq | Toshiba | IBM | Epson | Qume | NEC | Seagate | Wyse | Relysis | Zenith | Priam | Ranger

CIRCLE NO. 168

The *DESCRIPTION* section may finish with *EXAMPLES* of use and *DIAGNOSTICS* giving any error messages that the command may print. The MKS reference guide includes a useful *PORTABILITY* section at this point. This tells you what systems can run the MKS version of the command. The MKS toolkit runs on many systems, not just DOS. The section may also contain a reference to the UNIX variant of the command. It will show any extra features that may have been added to the MKS version to cope with the non-UNIX world.

Finally, the manual page ends with a *SEE ALSO* section pointing you at commands that relate to the manual page you are looking at. This is a good way of finding your way around the manual and discovering new commands: it's a printed hypertext link.

UNIX manuals are traditionally split into several numbered sections, and the MKS reference manual follows this lead. The largest section is Section 1 giving the commands that are supported by the system. The sections on system calls (2) and library routines (3) are empty since MKS does not provide a system call interface or a programming interface. Section 4 on file formats has several entries. The final section, Section 5, contains miscellaneous entries that gives DOS-specific information.

I give all this documentation full marks. It is accompanied by an on-line `help` command giving summary information. You type `help` followed by the name of the command in which you are interested This

***You can go the whole hog and login to your Dos machine just like UNIX***

is invaluable for those occasions where you know that a command does something but are unsure of the option needed to make it happen. It's also a boon when on a train and the manuals are sitting at home.

## Recommendation

You have probably gathered by now that I like this product. I now find myself writing shell scripts for DOS. If I need to invoke them from `COMMAND.COM`, I have a short DOS batch file that merely invokes the shell on the appropriate shell script. I have it installed on all the DOS machines in my immediate environment and switch into it whenever I intend to do any real work. My laptop has a minimal installation, just the shell and a few choice commands. This isn't just an old UNIX user refusing to change to DOS, I really do miss those UNIX tools and the way that they inter-operate to get the job done.

## Product info

The MKS toolkit is distributed in the UK by several people, at several prices: Software Construction Ltd, 0763 244114 quoted £165 + VAT; Grey Matter Ltd 0364 53499 said £169 and System Science 071 833 1022 offer the toolkit at £155. The prices are for the DOS version, expect to pay more for the OS/2 version and even more for some code to do both. The most recent release is version 3.1.

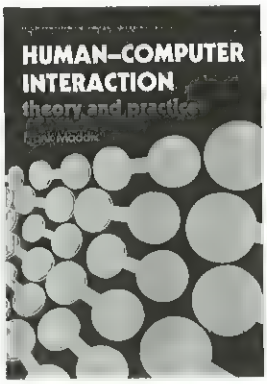
EXE

*Peter Collinson is a freelance consultant specialising in UNIX. He can be reached electronically as* `pc@hillside.co.uk` *(although your mailer might be happier to put the address the other way round) or by phone on 0227 761824.*

# Books

*Humans, Computers and Objects*

## HAL, where are you?



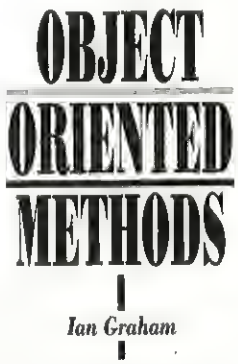

In under a decade it will be 2001. Do you remember the movie? Although our software is far more intelligent than it was 10 years ago, there is still a long way to go before we ever produce a computer as intelligent as HAL; *Human Computer Interaction* will steer you in the right direction, or so it claims. After introducing the concepts behind Human Computer Interaction (HCI) in chapter one, Frank Maddix introduces a rather simplified model in the second chapter. This is all very well; but other, more advanced models are also introduced here and these tend to have complicated diagrams which appear to be out-of-place in the text. There is a good section on information processing and gathering and you may find the discussion on *on-line monitoring* of particular interest if you need to determine the most often used functions in your application.

From chapter five, *Human Computer Interaction* starts discussing some of the real issues behind good HCI design including, some which we take for granted. The first victim is the QWERTY keyboard which is an example of negative HCI engineering as it was designed to impede the speed of a fast typist. Other input devices such as pointing devices are also given good coverage. Colour and monochrome monitors are next on the agenda and even an X-terminal attracts Maddix's wrath since its text fonts are difficult to read. The chapter on voice I/O pays lip service to an immense field of study but, for such a superficial overview, you may become bombarded with some of the more demanding techniques that are covered.

There are numerous interfaces available to the computer user and Maddix compares programming languages to Command Line Interfaces and also drawing, painting and music packages. Integrated development environments and debuggers are also covered so end users of such products (ie the target audience of this book) should pick up a few useful tips on assessing these tools subjectively. The sections on choosing a good icon, writing dialogues and selecting aesthetically pleasing colours are guaranteed to blow your mind (eg in a paperless office how would one relate to the *Filing Cabinet* icon?). Maddix also covers devices for the physically disabled, blind and deaf computer user and there is even a brief paragraph on direct brain input (the implications of which dwell mainly in the 'Twilight Zone'). *Human Computer Interaction* provides a lighthearted overview of a subject that will play an important rôle in software development of the future and provides an abundance of useful techniques that can be incorporated into current software to make applications more usable.

*Title: Human-Computer Interaction*
*Publisher Prentice-Hall*
*Pages:306*
*Author:Frank Maddix*
*Price: £18.95*
*ISBN: 0-13-446220-3*

## My cat is object-oriented




*Object Oriented Methods* by Ian Graham is a dense piece of literature. I mean that in the sense that almost every sentence is thought-provoking. Consequently it is not a source that is easily dipped into, but it is all the better for it - readers will benefit most from a study of the whole text. The book is aimed at programmers, designers, analysts, project managers and students. The nine chapters cover such things as basic concepts and benefits, object-oriented programming, database technology, design and analysis. Because it takes a language-independent approach to the themes covered it is more generally appealing than other books which tend to be language-biased. Each chapter starts with a cleverly chosen quote to whet the reader's appetite (cf the heading of this review), and closes with a summary of the main issues dealt with. There is a huge bibliography, a glossary of terms and comprehensive indexes by name and subject.

Mr Graham does have a tendency to veer off into rather esoteric discourses on the nature of objects and semantics, but the work is so well written that it is always interesting and in the end his ideas are related to real-world examples in the business and commercial sector. This is especially true in the section of the book dealing with *Identifying Objects* - in the analysis and design phase of a project. The pages are littered with references to the work of philosophers such as Hegel, Heidegger, Kant and Marx which makes for fascinating reading.

The book does not present object-orientation as a panacea and Mr Graham frequently points out that much of the technology (with regard to CASE, languages and databases) is still immature. He also recognises that there's no point in being an OOP purist when faced with the harsh reality of commercial business applications development, where compromises are inevitable. The author does however offer practical advice on how to use currently available object-oriented technology to achieve real benefits in terms of extensibility and reusability (at the programming *and* system specification level). There are also provocative musings on the future of IT in the wake of the object-oriented revolution.

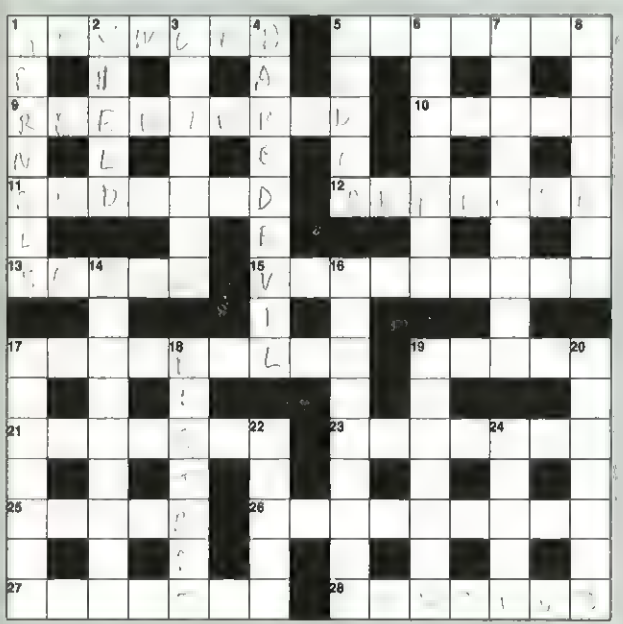
Many more topics are covered in this book than I have space to describe, including an appendix entitled 'Fuzzy objects: Inheritance under uncertainty' (with sub-section headings like *Defuzzification* and *Fuzzy objects, fuzzy quantifiers and non-monotonic logics)* which should not be attempted by those readers with a delicate constitution. *Object Oriented Methods* is an excellent book for its standard of writing and breadth of coverage and should prove an invaluable addition to the shelves of anyone considering or undertaking object-oriented development.

*Title: Object Oriented Methods*
*Publisher: Addison-Wesley*
*Pages: 410*
*Author: Ian Graham*
*Price: £19.95*
*ISBN: 0-201-56521-8*

## Books received this month

| | | | | |
|---|---|---|---|---|
| *A Primer for PHIGS* by F R A Hopgood and D A Duce | Wiley | £17.95 | ISBN:0-471-93043-1 | pp258 |
| *MsDos Programmer's Reference (incl Ver 5.0)* by Microsoft Corporation | Microsoft Press | £22.95 | ISBN:1-55615-329-5 | pp464 |
| *dBase IV beyond the Basics* by Mark Brownstein and Dan Gutierrez | Wiley | £22.95 | ISBN:0-47161-748-2 | pp458 |
| *80386/80486 Programming Guide* by Ross Nelson | Microsoft Press | £21.95 | ISBN:1-55615-343-0 | pp476 |

# SEPTEMBER .EXEWORD



## ACROSS

1 Reserved unit of language may unlock the system... (7)
5 ... and lock up the logic circuit (3,4)
9 Keyed in again to verify (9)
10 Negative is one way to produce interference (5)
11 French ten rude to former priest (2-5)
12 Violent rush through unit of volatile store (7)
13 Runs through loop up the ladder (5)
15 Very greedy for memory perhaps (9)
17 Pass below when too small to show (9)
19 Briefly I shall appear as a column in church (5)
21 Covered and ready to cook (7)
23 Against language law (7)
25 She's a new way to avoid tax (5)
26 Spread fewer rumours without 10 (9)
27 Timid cowboy? (7)
28 Columns between text columns (7)

## DOWN

1 Crunchy centres of systems (7)
2 Give up for final data ...(5)
3 ... and produces those data (7)
4 Valid deer chopped up by brave fool (4-5)
5 Snake in the circuit ... (5)
6 ... is lively and always changing (7)
7 They bring graphics to life (9)
8 Cricket teams almost ready for morning break (7)
14 As runs a rotten loop (9)
16 Starting to code again from scratch (9)
17 New versions may ape dust somehow (7)
18 RUN from line 1 once more (7)
19 Sounds like beer intended for one with a sickness (7)
20 Registers in two-way data structures (7)
22 Generous source of electrons (5)
24 Wild birds with half a horse in the home counties (5)

*'.EXEWORD' compiled by Eric Deeson*

**AUGUST .EXEWORD**

# Opportunities for Software Professionals

### SYSTEMS PROGRAMMER

**BERKSHIRE** **To £20K**

1-2 years COBOL experience within software tool development or applications programming. Knowledge of PC DOS and OS/2 with presentation manager. To provide product development and support skills for important COBOL tools.

*Ref: 01 - 8/91*

### SOFTWARE PROGRAMMERS & ANALYST AS400/ IBM - SYS 36/38

**MOST AREAS** **To £25K**

We have clients seeking experienced Software Engineers who have worked in an AS400/IBM mainframe environment. Positions range from programmers through to Senior Analysts and Support personnel.

*Ref: 02 - 8/91*

### SYSTEMS ENGINEER

**HAMPSHIRE** **£Neg**

To provide a problem solving function in development department and generate high integrity solutions to difficult problems. Minimum experience to include five years hardware/software problem solving, six months low level programming, knowledge of classical data structures, 1st Honours Degree, 22-30 years old

*Ref: 04 - 8/91*

### SENIOR SYSTEMS ENGINEER

**HERTFORDSHIRE** **£18-22K**

Our client is engaged in a continuing programme of development and support of microprocessor based real-time data acquisition and analysis systems of PC-based support packages. They are seeking an experienced Systems Engineer - in both hardware and software - and their combination into successfully integrated systems - to manage and take part in the engineering activities of several new projects. Must have an electronic engineering/scientific background and proven skills in project management.

*Ref: 05 - 8/91*

### TECHNICAL SUPPORT

**THAMES VALLEY** **£Neg**

Various clients are seeking young, enthusiastic, goal orientated personnel, willing to work in a fast moving environment. Must have two to three years experience in either LANS, Windows, UNIX, EXCEL with a Degree or equivalent.

*Ref: 06 - 8/91*

### SENIOR PROGRAMMER

**LONDON** **£17-23K**

1-3 years experience of at least 2 of either IBM P52/752, C programming language/windows, and experience in presentation manager Easel.

*Ref: 03 - 8/91*

### 68000 PROGRAMMER

**HAMPSHIRE** **£Neg**

Responsible to take high level descriptions of functional requirements, undertake the design and produce the design documentation, implement and test the solution. Must have 68000 Assembler programming experience (with some 'C') and have worked in a networked PCAT development environment. Minimum requirements include six months low level programming experience, low level programming of True Real Time tasks, 2:1 or 1st Honours Degree, 22-30 years old. *Ref: 07 - 8/91*

### SYSTEMS PROGRAMMER

**BERKSHIRE** **£13-15K**

To provide programming and maintenance support for end-user products in the IBM PC market. Experience of 8086/80286/ 80386, Assembler, DOS or OS/2 for creation of COBOL code generator. *Ref: 08 - 8/91*

### PROGRAMMER

**LONDON** **£16-20K**

Must have minimum of one years experience of Oracle including six months of Oracle* Forms and/or C programming language. Degree level qualification, PC programming experience and city-based applications. Responsible for writing programs, creating test schedules installation and diagnosing faults on site. *Ref: 09 - 8/91*

For further information on these or related positions call
JEREMY WILLAN or SARAH HOLTHAM on
(0734) 774234 or (0604) 33195 after 7p.m. FAX: (0734) 772773
Or write in confidence to CPS at:-

**63 Peach Street**
**Wokingham**
**Berkshire RG11 1XP**

CPS
COMPUTEC PERSONNEL SERVICES

# * ASSOCIATE CONSULTANTS *

- **OS/2 KERNEL**
- **OS/2 SUPPORT**
- **LANS/COMMS SUPPORT**
- **LANS/COMMS PROGRAMMING**

***We are QA Training Limited, a highly successful computer systems training and consultancy company. Formed in 1985, QA now has an enviable position as a world leader in computer systems training, with an international client base, plus the support of many manufacturers.***

***We are seeking to expand our team of self employed Associate Consultants, who could work with us on a number of projects relating to the OS/2 environment, and the LANS/COMMS area.***

## OS/2 SUPPORT:

You will have experience of working in a technical support centre environment, responsible for installations, back-ups, and fault finding. You should be able to demonstrate technical skills in:

- **batch file programming**
- **major software packages, such as Windows/PM environment**
- **languages, such as Basic, Pascal or C**

If you are also familiar with a DOS, UNIX or VAX environment, this would be particularly useful.

## OS/2 KERNEL:

Where you would be involved with projects based on OS/2 as a platform, and should therefore be able to demonstrate considerable experience in C and a working knowledge of OS/2 Kernel.

Additional skills, which would be preferable include:

- **MASM**
- **C++**
- **PM/Windows**
- **VMS/UNIX/DOS/IBM 370**

If you are also familiar with SQL Database Manager/SQL Server/IBM OS/2 extended edition, this would be particularly useful.

## LANS/COMMS SUPPORT:

If you can demonstrate expertise in any of the following areas, we would be pleased to hear from you:

- **Installation, configuring and tuning of IBM Lan Server**
- **Configuring IBM OS/2 Communications Manager, to include Terminal Emulation (3270, 5250); SNA Gateway; SNA Profiles including LU 6.2 and APPC**
- **Data Communications - WANs, ISDN, X25**
- **Familiarity with OS/2, DOS, UNIX operating systems**
- **Some familiarity with CICS configurations and systems programming**

If you also have experience of mainframe products such as NetView, this would be ideal.

## LANS/COMMS PROGRAMMING

You will have experience of working within the Data Communications environment, and will be able to demonstrate the base skills as above plus

- **CPI-C or APPC**
- **IBM Lan Server or MS LAN Manager**
- **Knowledge of other LAN environments such as DecNet would be particularly beneficial.**

## PM AND ADVANCED PM

We have an immediate need for OS/2 Consultants able to demonstrate a broad based experience, which includes:

- **PM Programming**
- **Advanced PM Programming**
- **Database Manager**

For any of the above opportunities, please send a detailed Curriculum Vitae to:

***Melinda Bishop, Personnel Manager, QA Training Ltd,***
***Cecily Hill Castle, Cirencester, Glos GL7 2EF***
***Tel: 0285 655888***

QA TRAINING

# REASONS TO BE CHEERFUL (PARTS 1-9)

Associated FT Personnel

- UNIX
- DATABASES
- 4GLs
- OPEN SYSTEMS

Evening Telephone:

Gerard Fawcitt
081-341 7301

Marcus Langford-Thomas
081-681 3905

Facsimile: 071-487 4501

telephone
071-487 4110

### E. MIDLANDS
### £ Negotiable - Good

Our client is an established Software House, with a network of offices in the UK. Their Cambs office wishes to recruit software professionals, from Programmers to Team Leader, to work on client-driven projects, and Bespoke software development for packages.
Development is carried out in Informix 4GL on UNIX-based Platforms, but the company would be happy to cross-train if you can demonstrate a good record of Software Development with another UNIX-based 4GL.
This is an outstanding opportunity to join a successful organisation which can offer a career path, including on-going Training.
*Please quote reference: G Hurst*

### SUSSEX
### £18 - 22,000 Negotiable

This organisation, a multi-national "household name" is currently seeking 2 Analyst/Programmers. Successful candidates will work on continuing Development of substantial Travel systems.
You will need to have 2 years UNIX/'C' Commercial experience, some background in Analysis, and Structured Methods.
These positions are excellent career opportunities.
*Please quote reference: M Peters*

### BERKS
### To £22,000 + Car

Our client develops and markets a range of Engineering Management Systems for the European market. These systems are technically highly complex, requiring a strong technical awareness.
To help them support these systems, they require a Technical Support Specialist with a systems knowledge - either UNIX or VMS - plus a strong LAN understanding particularly Ethernet, TCP/IP or DECNet, Token Ring.
A strong knowledge of 'C' is essential as the majority of problems will involve the company's source code
A foreign language such as French or German would also be highly desirable though not essential, as most clients are mainland Europe-based, although the majority of work will involve working from the company offices in Berkshire.
The company can offer excellent prospects of career development and really challenging Technical Support problems on an excellent product range.
*Please quote reference: R Charlton*

### HERTS
### £ Negotiable + Car

This represents the opportunity to work for the world's leading supplier of UNIX Systems, as the company's premier Network specialist.
The candidate for this position will possess mature and extensive problem-solving skills, being able to take an overall perspective on support issues. Support will be final line to all distributors, OEM's and developers.
Technically, you will need a minimum 8 years experience, having gained indepth software development (to source code level) understanding of LAN Manager, OSI products and TCP/IP with UNIX Kernel knowledge, particularly Sockets and Streams.
The work will require resolving complex problems, having overall support responsibility for key networking products, and liaising with Engineers from other departments as part of the development review team.
This is a significant and demanding role.
*Please quote reference: G Banks*

### C. LONDON
### To £30,000 + Full Banking Benefits

Our client is looking for a Senior Analyst/Programmer with a background in Financial Modelling experience eg Foreign Exchange, SWOPS Futures and Options; plus at least 3 years 'C' under UNIX development experience.
You will be working with a team including Mathematicians and Programmers, developing the latest models for financial markets. Experience of relational database (particularly Sybase) would be useful, though not essential.
The company can offer excellent training, career development and a good remuneration package.
*For further details please quote reference: N Stiles*

### MIDDLESEX
### £18 - 23,000

Specialists in environmental Control Systems and part of a successful International group, this organisation is committed to new product Research and Development.
As part of a new development phase, it now intends to recruit 4 experienced Software Engineers to join newly formed multi-discipline teams.
Projects will entail real-time, Data Handling and "Front-end" software development in a UNIX/Workstation environment.
If you can demonstrate 2-3 years software Design and development in relevant areas, coupled with a solid UNIX/'C' background, contact us right away.
*Please quote reference: G Cohen*

### MIDDLESEX
### To £23,000 + Bonus

We have an immediate requirement for an experienced UNIX Trainer. The brief, within this major product supplier, is to join a dedicated team that provides high quality Training on UNIX fundamentals and application software.
In a dynamic and fast-moving environment, you will be creating and delivering training courses of a high standard. The successful candidates will be prepared to gain a thorough understanding of the business environment. Essential is a minimum of 1 year's training and knowledge of UNIX (preferably to systems level), communications (LAN's/WAN's) and Office Automation Systems (eg Uniplex, Office Power).
*Please quote reference: R Wilson*

### C. LONDON
### to £25,000 + Bens.

Developing a new range of Financial terminals, this company has the backing of one of the world's largest corporations.
With Open Systems being the underlying technology, the company is looking to build a team of Systems Engineers with between 1 and 5 years experience of UNIX to at least Device Driver level; knowledge of 'C' or C++ is essential, with experience of 68000/88000 Assembler (or other Assembler) useful.
Knowledge of X-Windows at XLib level is also highly desirable. This is a demanding and extremely interesting "clean-sheet" development with the opportunity to work for a highly successful worldwide organisation.
*For further details quote reference: B Moore*

### S. BUCKS
### £19,000 + Benefits

Having established continued growth over the past 5 years (including the last year) our client seeks 3 Software Engineers for new positions.
Working on the continued improvements to a PCB layout system, you will develop meaningful specifications, and prototypes from such specs, and carry through development to a full working system.
A minimum of 3 years 'C' Development experience in a UNIX workstation environment, knowledge of graphical user interfaces (Open Look, Motif) is essential. A knowledge of Object Oriented Coding techniques and familiarity with PCB layout is useful though not essential.
You will be working in small teams and therefore must be a good team member, willing to contribute ideas.
*For further details quote reference: A Ball*

**FAWCITT THOMAS ASSOCIATES, 11 DUKE STREET, LONDON W1M 5RA**

---

## C, C++, UNIX CAREERS

**C++ Programmer £25k + travel SURREY**
Expanding international company seeks experienced C++ programmer to produce C++ CASE tools. In-depth C++, good software quality experience and use/knowledge of CASE tools essential. *Ref AC*

**C Programmer £18k+ MIDDLESEX**
2+ years commercial C software development? Knowledge of user interfacing and client contact? Real option to make a positive contribution to a dynamic company. *Ref SB*

**UNIX Designer £25k+ LONDON**
UNIX systems configuration experience? Interested in scientific applications to be ported from high speed processors to high performance workstations under AIX RS6000. *Ref ID*

**UNIX SPECIALIST £20-25k BRISTOL**
State-of-the-art manufacturer requires a UNIX guru with significant experience at Kernel level and writing Device Drivers. Mission is to port UNIX onto new hardware platforms. *Ref FD*

**For these and other vacancies, please contact:**
**Ian Dunn on 081 549 6441 (days) or 081 653 6443 (evgs)**

**ab executive**

59 Eden St, Kingston-Upon-Thames, Surrey KT1 1BW. Tel: 081-549 6441 (24hrs)

---

## *Chess Computer Services Ltd*

***Search, Selection and Advertising Consultants***

*Park House, Greenhill Crescent, Watford Business Park, Hertfordshire, WD1 8QU*
*Tel: 0923 225363 Fax: 0923 225051*

**LISTED IS A SAMPLE OF CURRENT VACANCIES**

**5 Vacancies - Senior UNIX Engineers - to £30k + Car**
Need a seasoned development engineer to work on a particular offering of UNIX With years of experience in UNIX you will have worked down to kernel level. Networking experience would be useful ie TCP/IP NFS etc
You will have a thorough knowledge of the 'C' programming language and be able to work unsupervised.

**INFORMIX Analyst Programmer - senior - to £30k**
To work on a major database - maintenance development and enhancement work - (excellent company benefits).

**UNIX/NETWORK SPECIALIST - TEAM LEADER - TO £35K + BENS**
Working to source code level for fault finding, this position is as technical authority on networking products.

**ORACLE ANALYST PROGRAMMERS - TO 28K**
To work in DEC-VMS environment within a financial institution.

**PARADOX - ANALYST PROGRAMMER - £20K**
Major retailer

**COBOL ANALYST PROGRAMMER - DEC/VMS - £25K**
Telecomms supplier (excellent management prospects)

**SENIOR C ASSEMBLER PROGRAMMER - £20K**
Network manufacturer

**IF YOU HAVE EXPERTISE APPLICABLE TO THE ABOVE OPPORTUNITIES OR YOU HAVE THE FOLLOWING EXPERIENCE**

## CALL NOW!

**UNIX - INFORMIX - LAN's - DATACOMMUNICATIONS**

**Tel: 0923 225363**
**Fax: 0923 225051**

**£18-£20,000** **SURREY/HANTS**

# 'C' AND 4GL PROGRAMMER
# Executive Information Systems (EIS)

## The Company

Backed by significant and secure financial resources, this recently formed company has developed the next generation of object-oriented EIS.

## The Role

- Work on leading edge developments and implementation issues in open systems, multi-media and graphical user interfaces
- Enjoy a dynamic working environment with talented individuals, and exploit your strong 'C' and UNIX programming skills.

## The Candidate

You have a minimum of 3 years' experience with 'C' and UNIX, and ideally some related skills including X-Windows, MS-Windows, SQL databases, 4GL's and graphics. You recognise the potential of career development within an exciting and rapidly expanding Company, and have the initiative to exploit that potential.

***Call Robert Hobbs on 0252 876520, or alternatively send him your c.v. at Caves Farm House, 33 High Street, Sandhurst, Surrey GU17 8EB. Fax: 0252 890028***

INPHASE
SOFTWARE LIMITED

---

### C/C++/Clipper/Oracle

**Surrey** **to £25K + Benefits**

Greenfield opportunities with this young, dynamic services company committed to building its efficiency through its IT skills and technology. Programmers are required with a minimum of 2 years experience with ORACLE or CLIPPER and C or C++. Analysts with 4 years experience using structured design and analysis with C/C++ and preferably ORACLE are also in demand. You will be working with the latest software tools and be involved in developing exciting new projects.

*Ref: CT*

### C/UNIX/Graphics/Windows

**South East** **to £25K**

Leading the development of computer based solutions, this young innovator is offering commercially minded software engineers the opportunity to further their career at the forefront of Graphics Technology. You will probably have 2 years+ experience of C/UNIX and possibly an appreciation of graphics, windows, VRTX, porting, RISC or 680*0, microprocessor hardware.

*Ref: TM*

### C++/C/OOD/Windows/SQL

**Herts/Cambs** **£15-22K**

Exciting new opportunities have been created by this successful software services specialist - seeking to build a new development team. You will be working with the latest tools including C++, OOD, SQL MS-Windows and structured methods. You will be educated to degree level with a minimum of 2-3 years experience with at least 2-3 years experience with at least two of the skills above - call now for further information.

*Ref: CT*

### GUI Specialists

**B'ham/Middx** **to £26k**

This rapidly expanding, successful communicatons company require experienced professionals developing GUI preferably using 'card' based MAC/PC tools. C and VAX VMS would be an advantage.

*Ref: CT*

### For more details

Contact Paul Innes (PI) or Teresa Maddern (TM) or Christine Trybus (CT) for details on the above and many other vacancies on

**0442 231691** days
or 0442 69740 eves/wkends
(fax 0442 230063).
Alternatively write to them at:

**Executive Recruitment Services, Hempstead House, Selden Hill, Hemel Hempstead HP2 4LT**

# RONICOM
RECRUITMENT

5-7 Sedley Place (off Oxford Street), London W1R 1HH
Tel: 071 491 3640 Fax: 071 499 2546

**BIOS SOFTWARE ENGINEER** up to £18k Surrey
Low level, Assembler, TASM, MASM, device drivers and a good mixture of Hardware and Software for this esteemed Manufacturer.

**SOFTWARE DEVELOPER** up to £20k Cambs
C, UNIX and X-Windows programmers required for a recognised Software House. 1-2 years experience, additional expertise on PCs or Workstations is an advantage.

**INGRES/ORACLE/SYBASE** up to £25k London
You will have had experience in large projects and SSADM within an established consultancy. Preferably Systems House experience.

**ORACLE/TECHNICAL SERVICES** up to £25k London
A Technical Support/Consultancy role, systems development, installations, on-site advising, in-house administration. Platforms are VMS/PCs/Networks.

**WINDOWS/C/SECURITIES** up to £22k London
Either PC or UNIX platform knowledge, database experience would be an advantage for this C/Windows development role in a Securities environment.

**PC/NOVELL NETWORK SKILLS** £18k + Car London
Energetic, Support Professional needed for running client site networks at this large Comms. House. Hardware and software ability essential.

**JUNIOR CODER** up to £14k Dorking
C - UNIX - Windows - Oracle. Mac Interfacing would also be useful. Commercial experience is a must.

**JUNIOR CODER** up to £15k Berkhampsteed
C - Windows - 4GL & SQL types of skills, one year experience upwards. Rewarding software project.....

**C SOFTWARE DEVELOPER** up to £18k London
C - Assembler - LAN - Real time and VGA card skills. Good hardware expertise and structured techniques would be a bonus.

**C LAN PROGRAMMER** up to 18k + Bonus SW London
Competent in DOS environment, OSI/ICL networking skills, X25.

**DOS/LAN SENIOR ANALYST PROGRAMMER** £21k + Bens Surrey
Fully competent in C - 3-4 years experience, network programming, data communications, and a few successful software implementations under your belt.

**SOFTWARE PRODUCT SUPPORT SPECIALIST** £18k + Car + Bens Berks
International support, providing solutions to queries, Assembler, Cobol, Workbench and MS-DOS and MVS.

**CHIP SET DESIGN CONTRACT** Good Sweden
ASICs, low level voltage, experience of vendor toolkits - Synopsis, Cadence, Varilog, Edge.

**OS2 TECHNICAL SUPPORT** £18k W. London
Must be experienced in Software House environment type products, Presentation Manager or Windows plus an understanding of 4GL applications.

**GRAPHICS SOFTWARE DESIGNER** up to £25k Herts
SPARC, UNIX, RS 6000 platforms. C programming for Real time, image manipulations, graphics routines, device drivers, simulation etc. X-Windows useful.

**4GL PROGRAMMER** £18k N. London
IS2 and C languages required for Development and Support of Financial systems.

**INFORMIX PROGRAMMER** £16k Berks
2 years experience of this 4GL for work on development of, among other things, a new membership package.

**PRODUCT EVALUATION** £18k Neg upwards Berks
Technical support background ideal for this position to evaluate and assess performance and compatibility of PC and related components.

**SUN SYSTEMS EXPERIENCE** £15-25k W. London
2 positions requiring this platform. Systems programmer for large end user, which includes some Systems Admin. Also Field Service/Tech Support role for Sun/SunOS environments.

**INGRES ANALYST/PROGRAMMER** up to £20k Surrey
2-3 years experience for work on large international projects.

**SOFTWARE DEVELOPER** up to £25k S London
C, PCs, UNIX. Previous experience in EIS systems a distinct advantage.

**UNIX SYSTEMS ADMINISTRATOR** £18k London
Previous experience of running in-house system for large end user would be nice.

**LOW LEVEL DEVELOPMENT** 20k N London
68000 Assembler, Chip Sets and Electronics development for large Manufacturer. Knowledge of Communications and Digital techniques is also desirable.

If any of these positions are at all suitable, or even if your skills are not quite what is required, do not hesitate to give either Mike Dearing or Heather Goldstraw a call on 071 491 3640 or after hours on 081 767 1003

PEOPLE IN UNIX*

GET INTO GEAR
PREPARE FOR THE UPTURN

*Unix is a trademark of AT&T

**ACTIS**
is seeking people with good technical experience of **OPEN SYSTEMS** which is still a growing sector of the IT market.

We have a number of instructions in the North and Midlands and we are looking for people with specific skills in:

**Networking and Communications,**
**Relational Database/SQL**
**4 GLs, CASE, IPSE**
**C and WINDOWS**
**Accounts/Commercial and Scientific Programming**
**Training and Documentation**
**Technical Sales**

We are seeking consultant, project management, development and support staff from graduate trainees to those with ten years experience.

To discuss these and other opportunities talk to **Honor Lindsey** on **(0204) 20200** or send your CV to **ACTIS RECRUITMENT**
**17 CHORLEY NEW RD, BOLTON BL1 4QR**

ACTIS
PEOPLE IN UNIX*

**BUCKS** £16k - £20k
Major development requires Software PC Development Engineer with a minimum of 2 years application programming experience on **MS-DOS** with **'C'** preferably with MS-Windows experience. Other useful experience: **Image Processing, VAX/VMS, UNIX** or **Networking.**

**N.E.** c£21k
Project Leader needed for major R&D organisation in the field of Software Engineering & Telecommunications for European collaborative research initiatives. Knowledge of development life cycle, project planning, **UNIX, 'C'** and **Windows.**

**E.ANGLIA** to £25k
Major R&D development program requires at least 2:1 graduates with experience in some of the following: **C, UNIX, Real-Time, Neural Networks, C++, Voice Recognition, Speech Synthesis, OOD, DSP, OSI/X.25.**

**HERTS** £ good
International developer of Integrated office software is keen to recruit Software professionals at all levels with most of the following skills: **'C', UNIX, X-Windows, Motif, MS-Windows** or **Porting.**

**CITY** £ Excellent
Developing substantial delivery systems for brokerage and dealing rooms, based on **OS/2, Presentation Manager** and **Token-ring LAN's.** Our client requires **OS/2 PM programming, OS/2 kernel programming** and **Microsoft C.** Any of the following are useful: **MS-Windows, NetBIOS, X.25, SDLC, Async, MS-DOS, PVCS, UNIX, Stratus VOS, Oracle, Sybase** or **Ingres.**

**HERTS** to £23k
Software systems design engineer required to design and develop Information Management Systems, if you have 4 years + experience under **UNIX** and **DOS** using **'C'** in a **Communications/OSI** environment. **OS/2** and/or **Networking** experience useful.

**CAMBS** £17k - £23k
Software Developers with 2 years or more strong **'C'** or **C++** experience under **DOS** or **UNIX** required for premier consultancy. Any **Sybase, Btrieve** or **CScape** useful.

**SURREY/LONDON BORDERS** £ excellent
International PC Software developer seeking bright young professionals with at least 2 years experience in **'C'** under **Windows, P.M.** or **OS/2.**

071-371 8496
ACUMEN SEARCH & SELECTION INTERNATIONAL
EBC House, 1a Ranelagh Gardens, London SW6 3PA Fax: 071-371 8502

ACUMEN
SEARCH & SELECTION INTERNATIONAL

# G.I.S. – The Technology of the 1990's

The U.S. National Science Foundation defines a Geographical Information System as "A computerised database management system for the capture, storage, retrieval, analysis and display of spatial, locationally-referenced data."

What they don't say is that G.I.S. incorporates almost all of the emerging enabling technologies including "Open Systems standards", OOP and OOD techniques and Hypermedia as well as more conventional RDBMS and 4GLs. G.I.S. is growing and your career could grow with it!!

### Oracle/Ingres Database Designer

**£24,000** **Hertfordshire**

This solutions vendor markets Arc/Info, the world's most successful G.I.S. tool-set. You will be developing new database software to enable their large customers to integrate multi-vendor DBMS into sophisticated corporate information systems. Experience of CASE tools is essential.

### Software Developer

**£20,000** **C.London**

A new appointment with a market leader in geo- demographics and market analysis applications. Your mission will be to develop new software modules for the UNIX market. Fluency in 'C' is essential, knowledge of 'C++', X-Standards and SunOS is an advantage.

### G.I.S. Customer Support

**£17,000 plus car** **Surrey & Cambridge**

Two new positions with leading G.I.S. vendors, both seeking good inter-personal skills and a year or more of experience in Utilities or Local Government applications. Technical skills should include UNIX, 'C' programming and a Macro command language.

### Young UNIX/G.I.S. Programmer

**£20,000** **Holland and Germany**

This U.S. Systems Integrator dominates the digital mapping industry. Your role will be to work alongside European customers in the utilities industry in order to develop new applications in 'C' and FORTRAN 77. Graphics and database experience is very desirable.

### Software Engineers

**£21,000** **Cambridge**

This 3 year old company is the G.I.S. industry's innovator. They have technology transfer agreements with academia enabling them to propagate research into 5GLs, Object-oriented techniques and version-managed database technology. Your skills must include 'C', 'C++', UNIX and graphics.

### Programmer/Analyst

**£16,000 to £18,000** **Thames Valley**

If you hold a good Computer Science degree qualification and you have worked in industry for at least a year then your skills could be cross-trained to this application of G.I.S. They use Sun Sparc workstations under SunOS and 'C' for development.

**To apply for one of these positions, or to learn more about G.I.S. contact ALAN CARNELL at Concurrent Appointments, the G.I.S. Recruitment Specialists.**

**Concurrent Appointments**
**Software Recruitment**

27 FIELD CLOSE
HARPENDEN
HERTS
Tel: 0582 712976
Fax: 0582 764858

---

# DEVELOPMENT SPECIALISTS

### *Real-Time UNIX & 'C'*

£18-24,000

Due to its continuing success, this leading supplier of real-time systems to financial institutions across the world is currently looking to expand its development division based in Central London. A number of positions are available for Software Engineers with a minimum of 2 years experience, as well as Consultants with over 4 years experience. It is essential that all candidates have experience of a real-time programming environment coupled with one of the following: UNIX, 'C' and X-Windows. All suitable candidates will also have exposure to the complete project lifecycle. Although not essential, any knowledge of the following would be advantageous: trading room systems, AIX, Ultrix and VMS. The company offers excellent career development, the opportunity for work overseas and a range of benefits on top of a competitive salary.

*Location: London* *Ref: PCEX11/1*

### *X-Windows Analyst Programmer*

£20-25,000

This represents a rare and interesting opportunity to join an emerging Software House, based in Surrey. They are looking for a dynamic Analyst Programmer, with commercial systems experience, to become involved with the development of a new financial product.

Your technical expertise must include solid UNIX/C and X-Windows development, and preferably some exposure to Motif and Open-Look. For the right person this will lead to unlimited career progression into client liaison, support and consultancy, but we would like to stress that the role will involve, initially, a high percentage of coding.

*Location: Surrey* *Ref: PCEX11/2*

### *C on MS-DOS Development*

to £20,000

Our client, based in Central London, is the Software House arm of a major UK services company. The successful applicant will join a small and highly motivated team, developing dBase-compatible, multi-user database systems, complete with GUI/Windows "front-ends".

You should have a minimum of 18 months 'C', ideally Microsoft 'C' on DOS, and must be able to develop systems in a structured fashion. Whilst a degree is not essential, candidates should be able to demonstrate an 'A' level education. Any experience of Windows V.3 and The Software Developers Kit (SDK) would be advantageous.

*Location: Central London* *Ref: PCEX11/3*

### *Windows & 'C' Developers*

to £20,000 + Benefits

With at least 2years commercial experience in Windows and 'C' development, including at least six months Windows 3 experience, you could find yourself working on any one of several current Windows projects. Educated up to Graduate level you will be instrumental in implementing Windows based technologies into forward thinking companies. Financial experience would be an advantage, although opportunities exist in a number of development environments including end-users, financial institutions and software houses. Performance related bonuses mean that your skills and effort are fully appreciated, and competitive salary packages are on offer.

*Location: The City and Home Counties* *Ref: PCEX11/4*

For further details of these and other development positions, either **contract** or **permanent** please contact Conrad Hills, quoting the relevant reference, on 071-734 4010 (office hours) or 081-542 8724 (evenings/weekends). Alternatively, write to McGregor Boyall, Lyndale House, 49-50 Great Marlborough Street, London W1V 1DB or fax your CV on 071-734 1297.

**mcgregor boyall**
IT HUMAN RESOURCING

# ADVERTISERS INDEX



# STOB - Our Town

*Ms Stob takes stock of her surroundings.*

Our town. You'd love our town. Here is Basingbrack, bordered by the busy M4 to the north and the noble River Brackish, sliding peacefully through its concrete culvert like an eel on a corrugated roof, to the south. The glow of dawn is reflected in the tinted glass surface of the Tamp-On building, concealing the rude word which naughty Tamp-On Ltd cleaning persons have contrived to display on its north side, by leaving the lights on in carefully selected offices.

To the east, Dead Elm Avenue leads to the Light Industrial Pollution Estate, which is where I live. There was not enough money for Tarmac roads around here - the wise council instead chose to concentrate its resources on the Basingbrack central car park and council member nuclear bunker complex, which is the finest building of its type in the whole of Berkshire - so they used concrete instead, and frosted it over with a layer of Tarmac as thin as icing sugar on a wedding cake. Now, wherever a car wheel may roll, the marzipan is showing through, producing an effect like the negative print of a snow thaw.

But enough of the fringes, for the centre is the place to be. I bet there isn't a High Street in the whole world with a larger selection of banks, building societies and estate agents. Admittedly, the only supermarket is located in a field, 10 miles away, on the other side of the motorway; but anyway, what man would cook his own food when we have, right here in Basingbrack, a choice of Macdonald's, Burger King *and* Kentucky Fried Chicken?

What? Yes, there certainly is a lot of IT here. But I'd hate to give the impression that this is a one-profession town. Oh no. We're not all C programmers here. There's a little MOD-type establishment on Rural Smell Rd, where they have *dozens* of Ada programmers; and even a British Airways office which, it is rumoured, contains one or two APL types (although we wouldn't mix with *them*, of course; no right-thinking C programmer would deal with people whose language looks like Asterix swearing*). So you see we're very cosmopolitan.

**This joke © D O'Brien 1991.*

Social life? Loads. Mostly it's organised by pubs. There's the *George and Coprocessor*, down by the river, where all the Trekees hang out; and uptown there's the more traditional *Dog and Dongle*, favoured by the Tolkienists. Some weekends, it's most exciting, they have a big fight; when the regulars of the D&D come streaming down Nitrated Water St, waving their plastic orc-despatchers, to be met by a hale of phasor fire from the G&Cs. The whole thing is carried out in the spirit of good fun, though. After they've broken a few windows, they all get together to sing a few raunchy programmer's songs:

*He tried to get off with the analyst's daughter*
*But he only made her laugh.*
*He told her his floppy was five-and-a-quarter*
*When she knew it was three-and-a-half.*

And then there's sport. Every fortnight they hold the pin-ball league... What was that? Oh yes, well the reason I'm ringing the Samaritans...

INNOVATION
COMPUTER
CONSULTANTS

Despite the current recession, innovation are currently increasing their market share and expanding a reputation as a complete and more professional consultancy covering all aspects of permanent recruitment in the UK personal computer marketplace. Our 'ex-industry' and technical consultants ensure a complete advisory service is provided to both clients and applicants alike.

We have a large number of outstanding opportunities with Banks, Securities Houses, Insurance Companies and a wide range of other Blue Chip end-users, as well as a great many openings with Dealers, Manufacturers, Suppliers, Maintenance Companies, and Planning and Consultancy Centres.

At Innovation we relish the challenge of demonstrating the way forward in personal computer recruitment.

# UNIX/'C'
## Up to £23k + Benefits

Exciting opportunities for young and imaginative Software Engineers, with the ability to offer our client 2+ years programming experience using 'C' and should be thoroughly conversant with SQL, UNIX operating systems and PC technology. You will be working in a primarily in-house development role, although some on-site work will be involved in this highly professional and respected company.

# CLIPPER ANALYST PROGRAMMERS
## £16-25k + BB

This prestigious banking organisation is currently seeking top quality development professionals with CLIPPER skills. Applications will be considered from Analyst/ Programmers, Senior Analyst/Programmers and Team Leaders as there are vacancies at all levels. You will be given the opportunity to work on a variety of exciting front and back office development projects within a dynamic and professional environment. Successful applicants should have excellent interpersonal and team skills.

# WINDOWS & 'C' DEVELOPER
## £20-25K + BB

Our client, a city based bank, is at present seeking a young, dynamic developer with a minimum 2 years experience in Windows and 'C' development, including at least 6 months Windows 3 experience. Educated preferably to degree level. Financial experience advantageous. Candidates with a high degree of professionalism with a desire to succeed need only apply.

## OTHER OPPORTUNITIES

**'C' ASSEMBLER.** London £NEG Grad Calibre. C++ advantageous
**FOCUS.** City. Up to £30k. 2yrs Focus on M/F or PC. Oracle, Paradox or UNIX experience desirable.
**'C' & UNIX.**Surrey. Up to £25k. Grad. Computer graphics exp. advantageous.
**MICROSOFT 'C' A/P**. Essex. £NEG Up to V.6 with DOS. Application prog. essential.

*FOR FURTHER DETAILS OF THESE AND MANY MORE POSITIONS, PLEASE CONTACT:*
**CARL SMITH, 17/18 CLERE STREET, LONDON EC2A 4LJ. TEL: 071-253 0227 FAX: 071-253 0339**

# They say the first Fortran 90 Compiler won't be available until at least 1992.


# We say they're wrong.

NAG is pleased to announce the release of the world's first Fortran 90 Compiler – one of several exciting new products to be released in the NAGWare Service.

Fortran is still one of one of the most widely used scientific and engineering programming languages and the f90 Compiler will ensure it stays so. The NAGWare f90 Compiler is a full implementation of the ISO Fortran 90 Standard and is ready to meet the most demanding test of all – you.

***The future is in the present.***

NAGWARE™

***For further information about NAGWare and the other products and services available from NAG, please contact:***

NAG Ltd, Wilkinson House, Jordan Hill Road, OXFORD, OX2 8DR, UK, Tel: +44 865 511245 Fax: +44 865 310139 Telex: 83354 NAG UK G
NAG Inc, 1400 Opus Place, Suite 200, Downers Grove, IL 60515-5702, USA, Tel: +1 708 971 2337 Fax: +1 708 971 2706
NAG GmbH, Schleißheimerstr. 5, D-8046 Garching bei München, Deutschland, Tel: +49 89 3207395 Fax: +49 89 3207396

CIRCLE NO. 171